DISTANCES ON RANKINGS:
FROM SOCIAL CHOICE TO FLASH MEMORIES

BY

FARZAD HASSANZADEH

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Associate Professor Olgica Milenkovic, Chair
Professor Bruce Hajek
Associate Professor Sariel Har-Peled
Assistant Professor Narayana P. Santhanam, University of Hawaii
Professor Rayadurgam Srikant

Abstract

From social choice to statistics to coding theory, rankings are found to be a useful vehicle for storing and presenting information in modern data systems. Often, in order to process the information, an appropriately defined distance on rankings is required or at least helpful. For example, in social choice, the quality of the results of distance-based voting rules depends almost entirely on the chosen distance function; in statistics, distances between rankings are used to measure correlation and to model data; and in coding theory, in the context of rank modulation, designing codes with a given minimum distance is required for preserving data integrity.

It is however well-known that conventional distances are not adequate for many applications. Motivated by several problems from different disciplines, including problems in social choice and bioinformatics, we axiomatically introduce two novel classes of distances on rankings to address the shortcomings of conventional distances. In addition, we propose several algorithms for computing or approximating these distances. The algorithms are based on graph-search techniques, Viterbi-type algorithms, and dynamic programming. Furthermore, we present algorithms for rank aggregation using the proposed distances. One algorithm is based on finding a minimum weight bipartite matching and another is a PageRank-type algorithm in which the transition probabilities of a Markov chain model yield the aggregate ranking.

In the context of coding theory, we introduce permutation codes in the Ulam metric that were not previously reported in the literature. Compared to known codes, the proposed codes can handle more diverse types of errors, including arbitrary charge-drop errors as well as other errors that affect a single cell, such as read disturb or write disturb errors. We present capacity achieving codes that can correct a constant number of errors and asymptotically good codes that can correct a linear number of errors in the length of the code. Our results also include derivation of the asymptotic capacity of permutation codes in the Ulam metric and simple decoding methods for one class of constructed codes. As part of our exposition, we also highlight the close connections between the new code family and permutations with short common subsequences, deletion and insertion error-correcting codes, and substitution error-correcting codes.

To my parents.

Acknowledgments

Foremost, I would like to thank my advisor Prof. O. Milenkovic for her continued support and encouragement through my Ph.D. Her help and guidance have been invaluable to me and instrumental in the completion of this dissertation.

My sincere thanks also go to the other members of my doctoral committee: Prof. B. Hajek, Prof. S. Har-Peled, Prof. P. N. Santhanam, and Prof. R. Srikant. Throughout the years, I have benefited greatly from their help and advice and have learned a great deal from them.

My fiancee and my best friend, Homa, has been a source of faithful support, immeasurable kindness, and vital strength to me. I would like to express my most heart-felt gratitude to her.

Last but not least, I am forever indebted to my parents, Abbas and Maryam, to whom this dissertation is dedicated, and my siblings, Saina, Behzad, and Farnaz, for their unconditional love and unwavering support.

Contents

Summary of Notation

Notation	Meaning
n	length of permutation or ranking
$[n]$	$\{1, 2, \ldots, n\}$ for an integer n
$\mathbb{S}_n$	set of permutations of $[n]$
$\langle a_1 \cdots a_k \rangle$	a cycle of a permutation π where $\pi(a_i) = a_{i+1}$
$\langle a\ b \rangle$	a transposition, a cycle of length 2 for $a, b \in [n]$
$\phi(i, j)$	a translocation with elements i and j for $i, j \in [n]$
$\lvert s \rvert$	length of a sequence or a vector s
$a <_\pi b$	$\pi^{-1}(a) < \pi^{-1}(b)$ for $\pi \in \mathbb{S}_n$
$V(G)$	vertex set of a graph G
$E(G)$	edge set of a graph G
$G[S]$	subgraph of G induced by S for $S \subseteq V(G)$
(uv)	an edge in a graph G for $u, v \in V(G)$
$G - t$	a graph G with an edge $t \in E(G)$ deleted
$G - v$	a graph G with a vertex $v \in V(G)$ deleted
$\deg_G(v), \deg(v)$	degree of v in a graph G
$\mathbb{R}_{\geq 0}$	set of nonnegative real numbers
$\mathrm{lcs}(u, v)$	length of the longest common subsequence of u and v
$\mathbb{T}_n$	transpositions in $\mathbb{S}_n$
$\mathbb{T}(\pi, \sigma)$	set of transposition transforms converting π to σ
$\mathbb{M}(\pi, \sigma)$	set of transforms in $\mathbb{T}(\pi, \sigma)$ of minimum length
$\mathbb{A}_n$	set of adjacent transpositions in $\mathbb{S}_n$
$\mathbb{A}(\pi, \sigma)$	set of adjacent-transposition transforms converting π to σ
$\mathbb{L}(\pi, \sigma)$	set of transforms in $\mathbb{A}(\pi, \sigma)$ that pass through a line
$I(\pi, \sigma)$	$\left\{\{a, b\} : \pi^{-1}(a) < \pi^{-1}(b), \sigma^{-1}(b) < \sigma^{-1}(a)\right\}$
d	a generic distance
d_K	Kendall τ distance
d_T	transposition distance (Cayley distance)
d_H	Hamming distance
d_F	Spearman's footrule
d_R	Spearman's rank correlation
d_U	Ulam distance
φ	a generic weight function
d_φ	weighted Kendall distance for $\varphi : \mathbb{A}_n \to \mathbb{R}_{\geq 0}$
d_φ	weighted transposition distance for $\varphi : \mathbb{T}_n \to \mathbb{R}_{\geq 0}$

Chapter 1

Introduction

1.1 Motivation

A ranking of a set of candidates is an arrangement of the candidates according to preference or some other criterion. Rankings are a non-traditional data format that is receiving increased attention as a method of storing and presenting information due to the fact that comparing the degrees to which two objects possess a certain quality is far easier than assigning a number to represent them. Indeed, in many cases the latter is not at all possible. For example, it is more difficult to measure how much one liked a movie than to decide which movie one liked best when presented with two choices. Even if we assign numbers to movies based on how much we liked them, in other words, rate the movies, the ratings cannot be treated as measurements. The only valid operation that can be performed on the ratings is comparison; it is wrong to assume a movie with a rating of 4 is "liked twice as much" as a movie with a rating of 2.

In everyday lives we encounter rankings of many different types and in many different circumstances: We may rank sport teams based on performance [1], rank movies based on how much we like them, rank politicians based on how much we dislike them, or even rank pets based on how cute they look [2, 3]. We use rankings to decide which school to apply to [4], which car to buy [5], or which company to work for [6]. Not only are there popular websites dedicated to top-ten lists [7], there are top-ten lists of top-ten lists [8].

Rankings and the related theoretical tools are also common in many academic fields. In statistics, Kendall studied measures of correlation between rankings, with applications to psychology, education, and economics [9] and Marden described models and techniques for analyzing data in the form of rankings [10]. Rankings are used in many statistical tests, including the Kruskal-Wallis test [11], for deciding whether several samples come from a single population or from several different populations; and Friedman's test for identifying factors that cause significant variation in the variable under study. In computer science, rankings are used in designing and analyzing search engine and meta-search engine algorithms [12].

In the social sciences, including political science and economics, rankings are used as a tool to study individual and collective decision making [13–16]. Col-

lective decision making, often referred to as social choice, is concerned with the study of voting rules. In one scenario, each voter expresses her preference in the form of a ranking of candidates. A voting rule is used to determine a ranking as a representative of the set of votes provided by the voters. This process is called rank aggregation. Rank aggregation is also useful for condensing information that is presented in the form of rankings. For example, consider rankings of schools based on different criteria such as graduation rate and average class size, or rankings of a certain product produced by different experts. It is often useful to produce a single ranking from the given set of rankings.

Rankings are also used in artificial intelligence for the study of automated systems with many agents [17] as well as in recommender systems in the context of collaborative filtering [18]. In coding theory, transmitting rankings instead of absolute values was proposed to combat noise in power-line communication systems [19]. Furthermore, encoding data as rankings was also proposed for storage applications, such as flash memories, in order to facilitate writing into memory as well as reducing the effects of noise [20, 21].

In many applications of rankings, including most applications mentioned above, a notion of distance between rankings is required or helpful. For example, correlation between rankings can be quantified via distance measures as studied by Kendall [9]. In collaborative filtering, if user preferences are presented as rankings, distance measures between rankings can be used to identify users with similar tastes. This similarity information can then be used to suggest a product to a user based on the preferences of similar users.

One common approach to rank aggregation is distance-based rank aggregation, first proposed by Kemeny [15, 16]. In this approach, one finds the ranking that has the smallest cumulative distance to the set of votes. The distance used in distance-based rank aggregation fully determines the outcome and thus should be carefully chosen.

In coding theoretic applications of rankings, one needs to find a code consisting of rankings such that one ranking in the code is not converted easily into another by noise. Here, noise manifests itself by reordering entries. In order to be able to design good error-correcting codes, the distance with respect to which the code is designed must reflect common errors caused by noise.

The contributions of this work are threefold: First, we propose two new families of distances on rankings motivated by diverse applications of rank aggregation, including bioinformatics. Second, we describe how these distances may be computed or approximated and present a number of algorithms for rank aggregation based on the proposed distances. Third, we propose several code families, in a distance not previously used for rank coding in flash memories, with improved error-correcting capabilities and present decoding algorithms.

1.1.1 Rank Aggregation

Our exposition on rank aggregation is motivated by the observation that none of the conventional distance measures are flexible enough to adequately reflect factors that are important for measuring closeness of two rankings.

The first and the most commonly used distance for rank aggregation is the Kendall τ distance, introduced by Kendall in 1948 [9]. Mathematically, rankings can be modeled as permutations. The Kendall τ distance between two permutations is the minimum number of swaps of adjacent elements need to take one permutation to the other. This distance was rediscovered by Kemeny when he proposed distance-based rank aggregation in 1959 [15, 16]. Kemeny proposed a set of reasonable axioms for a distance function between rankings and obtained a unique distance that satisfied those axioms – the Kendall τ distance. Distance-based rank aggregation using the Kendall τ distance is commonly known as the Kemeny aggregation or Kemeny's rule. In 1978, Young and Levenglick [22] showed that the Kemeny aggregation is the unique aggregation method that has three desirable properties, namely, the Condorcet property, consistency, and neutrality [22]. As a result, the Kemeny aggregation is also referred to as the Kemeny-Young method.

Although the Kendall τ and many other distances including Spearman's Footrule [23] are well-established and well-studied, they cannot take into account two important factors. First, in many applications certain positions in rankings are more relevant than others, and thus changes to these positions must induce a larger distance [24–29]. For example, search engine users tend to click more frequently on the top-ranked results than lower-ranked ones [30] and thus it is more important for search engines to get the top-ranked results right than the low-ranked entries [26]. In such a setting, for instance, it is reasonable to require that the distance between rankings $(1, 2, 3, 4, 5, 6)$ and $(\mathbf{2}, \mathbf{1}, 3, 4, 5, 6)$ be larger than the distance between $(1, 2, 3, 4, 5, 6)$ and $(1, 2, 3, \mathbf{5}, \mathbf{4}, 6)$.

If admission judges have to decide which few candidates to eliminate, the bottom portions of their rankings are more relevant [31]. In predicting the results in sport league systems with promotion and relegation, such as the Italian and the English soccer league systems, correct prediction of a certain number of top-ranked and a certain number of bottom-ranked teams is more important than predicting the correct positions of mid-ranked teams.

Second, when swapping elements of a ranking, swapping elements that are similar, such as swapping two similar movies or swapping two job candidates with similar sets of skills, must contribute less to the distance than swapping dissimilar elements. When rank aggregation is used in such applications, new distance measures that can take positional relevance and similarity information into account are needed.

In order to address these problems, we propose two novel distance measures. The basis of our approach is Kemeny's set of axioms for distance measures on

permutations. By relaxing these axioms we arrive at classes of distance functions where different swaps are assigned different weights. We refer to these distance functions as weighted distances.

In addition to rank aggregation, weighted distances are useful for collaborative filtering [18]. In that context, by assigning larger weights to swaps at the top of preference rankings or by assigning smaller weights to swaps of similar candidates, one can define weighted distances that can be used to identify similar users more accurately.

Weighted distances can also be used in ordinal genomic data fusion. Since genomic data such as gene expression levels are highly dependent on the experimental conditions, quantitative fusion of data generated by different laboratories represent difficult tasks [32]. But for identifying genes involved in a certain disease, if genes were first ranked according to different features, and only then used in data fusion [33], a number of incompatibility issues could be avoided. This kind of rank transform [34] may be viewed as a process analogous to dimensionality reduction, and in genomic data fusion, it can serve the purpose of removing arbitrary and meaningless information in numerical data that only carry ordinal information. Weighted distances that assign large weights to top of the rankings may be useful in this context due to the fact that higher ranks are more likely to be less noisy than lower ranks.

1.1.2 Rank Modulation Coding

Flash memories consist of cells that can store charge levels. Due to aging, environmental conditions, and design, the cells are subjected to charge leakage, leading to the phenomena of low memory endurance [35]. Due to this leakage, the information content of the cells may be compromised beyond the possibility of correction. Furthermore, in flash memories, while the charge of individual cells may easily be increased, it is difficult to reduce the charge of a single cell without affecting its neighbors [21]. As a result, to reduce the charge of a given cell, one needs to erase a block containing several cells. The difficulty of reducing the charge of a single cell is particularly problematic when writing to a cell: If the charge placed on a cell is more than the intended amount, that is, if an overshoot occurs, a block of cells needs to be erased and data needs to be rewritten on all cells in the block.

To combat the effects of charge leakage and to avoid unnecessary block erasures, Jiang et al. [20, 21] proposed using rank modulation in flash memories. The main idea is to store information as a ranking of cell charges rather than absolute values. For example, to store the ranking $(3, 1, 4, 2)$, cell 3 is programmed with the largest charge, cell 1 with the second largest charge, and so on. Using rank modulation, along with a cell programming method called "push to the top" [21], alleviates the overshoot problem mentioned above. Additionally, a storage error occurs only if a charge of higher level leaks below the level of a

previously less charged cell. If the leakage rate is small and almost the same for all cells, the ranking represented by charge levels is not affected by leakage in short periods of time. This reduces the number of errors caused by charge leakage.

Most leakage models assume somewhat uniform relative cell charge leakage, allowing only for the possibility of exchanging the ranks of two consecutive symbols. In this case, the distance function for input-output sequences of flash memories is the Kendall τ distance between permutations [20, 21, 36, 37].

However, if the number of possible charge levels is large, and a cell for a variety of reasons has a higher leakage rate than other cells, the charge of this cell may drop below the charge of a large number of other cells. While such an error may be modeled as a sequence of adjacent swaps, it should be noted that this type of error is the result of a single error event, and so it should be modeled as a single error. For this reason, we argue that a different distance measure should be used, namely the Ulam distance [38], which is the minimum number of elements that need to be "moved" to change one permutation into another. Furthermore, the Ulam distance is well suited for modeling errors that are not caused by leakage, such as read disturb and write disturb errors [39].

In our exposition on permutation codes for flash memories, we find the asymptotic capacity of codes in the Ulam metric and propose several families of codes in this metric, including codes with asymptotic rate 1 that correct a constant number of errors, as well as asymptotically good codes. Furthermore, we present a decoding algorithm for one class of the proposed codes.

1.2 Structure of Thesis and Summary of Results

In this section, we present the outline of the dissertation and summarize our results.

In Chapter 2, we introduce a new type of distance on rankings called the *weighted Kendall distance*, which is the minimum weight of a sequence of swaps of adjacent elements that takes one ranking to another, where each adjacent swap is assigned a nonnegative weight. We show that the weighted Kendall distance is the unique solution to a relaxed set of Kemeny's axioms. We present algorithms for computing or approximating the weighted Kendall distance. While a polynomial-time algorithm for computing the weighted Kendall distance for an arbitrary set of weights is not known, the weighted Kendall distance can be computed efficiently when weight are decreasing, that is, when the weight of swapping candidates at higher ranks is larger than the weights of swapping candidates at lower ranks. Decreasing weights represent an important case as such weights result in distances that are more sensitive to changes at the top of rankings than changes at lower positions. We also show that the distance can be approximated efficiently within a factor of two for any set of weights.

Chapter 3 is devoted to the *weighted transposition distance*. The weighted transposition distance, the second distance that we propose, is the minimum weight of a sequence of swaps that take one permutation to another when each swap is assigned a nonnegative weight. Note that swaps of non-adjacent elements are also allowed. If all weights equal one, the weighted transposition distance reduces to the Cayley distance, the minimum number of swaps needed to convert one permutation to another. A simple result, established by Cayley in 1849 [40], indicates that the Cayley distance can be computed in linear time. Computing the weighted transposition distance is however substantially more challenging. Although at this point it is not known if the problem is NP-hard, we show that large families of weight functions – such as weights based on metric-paths – have exact polynomial-time algorithms. Furthermore, we devise algorithms for approximating the weighted transposition distance for arbitrary weights, with an approximation constant that does not exceed 4. These algorithms represent an amalgamation of well-known algorithms in coding and computer science, such as the Viterbi algorithm, the Bellman-Ford shortest path algorithm, and dynamic programming methods.

In Chapter 4, we discuss rank aggregation algorithms for weighted distances. The problem of rank aggregation is known to be NP-hard for the Kendall τ distance [12]. The same holds for weighted distances as well. We present two approximation algorithms for rank aggregation with weighted distances based on known approximation algorithms for rank aggregation with the Kendall τ distance. One algorithm is based on minimum weight bipartite matching and produces a 2-approximation, while the other is based on a Markov chain model with candidates as states and transition probabilities that are determined by the set of votes.

In Chapter 5, we propose using the Ulam distance on permutations to model the effect of errors in flash memories. We study the relationships of codes in the Ulam distance with codes in other common metrics and present the coding capacity in different metrics. We design asymptotically good permutation codes in the Ulam distance by interleaving codes in the Hamming metric and present decoding algorithms for one class of constructed codes.

1.3 Notation, Definitions, and Preliminaries

In what follows, we present the definitions used in the subsequent chapters. We also present some simple results regarding the relationships between different distances on permutations. A summary of notation is given in the back matter for quick reference.

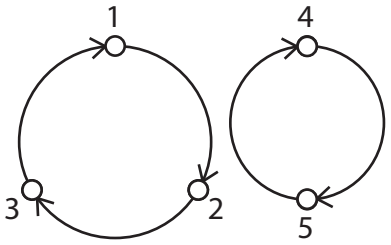


Figure 1.1: The digraph $\mathrm{dig}(\pi)$ of the permutation $\pi = \langle 1\ 2\ 3\rangle\langle 4\ 5\rangle$.

1.3.1 Definitions Regarding Permutations

Let $[n] = \{1, \ldots, n\}$. A *permutation* of $[n]$ is a bijection from $[n]$ to itself. The set of permutations of $[n]$ is denoted by $\mathbb{S}_n$. A permutation $\pi \in \mathbb{S}_n$ may be written as $(\pi(1), \ldots, \pi(n))$. The identity permutation $(1, \ldots, n)$ is denoted by e and the inverse of a permutation π is denoted by π^{-1}.

The product $\pi_2\pi_1$ of two permutations π_1 and π_2 is the permutation obtained by first applying π_1 and then π_2, i.e., $(\pi_2\pi_1)(i) = \pi_2(\pi_1(i))$ for $i \in [n]$.

The *functional digraph* of a function $f : [n] \to [n]$, denoted by $\mathrm{dig}(f)$, is a directed graph with vertex set $[n]$ and an edge from i to $f(i)$ for each $i \in [n]$. We use the words vertex and element interchangeably.

For a permutation π of $[n]$, $\mathrm{dig}(\pi)$ is a collection of disjoint cycles since the in-degree and out-degree of each vertex is exactly one. The *cycles of a permutation* are the cycles of its functional digraph. A cycle of the permutation may be written as $\langle a_1 \cdots a_k\rangle$, where k is the length of the cycle and there is an edge from a_i to a_{i+1} in the functional digraph for $i \in [k]$, with the indices evaluated modulo k.

With slight abuse of notation, a cycle σ is also used to refer to the permutation that has σ as a cycle and has all other elements as fixed points. Thus, the product of cycles is well-defined: the product of two cycles σ_1 and σ_2 is the product of the corresponding permutations. For example, we have $\langle 1\ 4\ 2\rangle\langle 2\ 5\rangle = \langle 2\ 5\ 1\ 4\rangle$. In particular, we can write every permutation as a product of its cycles. This represents the *cycle representation* of a permutation. For example, as shown in Figure 1.1, the cycle representation of $\pi = (2, 3, 1, 5, 4)$ is $\langle 1\ 2\ 3\rangle\langle 4\ 5\rangle$ and thus we may write $\pi = (2, 3, 1, 5, 4) = \langle 1\ 2\ 3\rangle\langle 4\ 5\rangle$.

A cycle of length two is a *transposition*. A transposition with elements a and b is denoted by $\langle a\ b\rangle$. The set of all transpositions in $\mathbb{S}_n$ is denoted by $\mathbb{T}_n$. An *adjacent transposition* is a transposition $\langle a\ b\rangle$ such that $|a - b| = 1$. The set of adjacent transpositions of $\mathbb{S}_n$ is denoted by $\mathbb{A}_n$.

Note that for $\pi \in \mathbb{S}_n$, $\pi\langle a\ b\rangle$ is obtained by swapping elements in positions a and b in π, and $\langle a\ b\rangle\pi$ is obtained by swapping a and b in π. For example, $(3, 1, 4, 2)\langle 2\ 3\rangle = (3, 4, 1, 2)$ and $\langle 2\ 3\rangle(3, 1, 4, 2) = (2, 1, 4, 3)$. For more details and examples, see §3.2.1.

For distinct $i, j \in [n]$, a *translocation* $\phi(i,j)$ is a permutation obtained from the identity by moving i to the position of j and shifting elements between i and j, including j, by one:

$$\phi(i,j) = \begin{cases} (1,\ldots,i-1,i+1,i+2,\ldots,j,i,j+1,\ldots,n), & \text{if } i < j, \\ (1,\ldots,j-1,i,j,j+1,\ldots,i-1,i+1,\ldots,n), & \text{if } i > j. \end{cases}$$

As an example, let $\sigma = (1,3,5,7,2,4,6,8)$. We have

$$\sigma\phi(3,6) = (1,3,7,2,4,5,6,8),$$
$$\sigma\phi(5,2) = (1,2,3,5,7,4,6,8).$$

For $i < j$, the translocation $\phi(i,j)$ is called a *right-translocation*, and the translocation $\phi(j,i)$ is called a *left-translocation*. Observe that the inverse of the left-translocation $\phi(i,j)$ is the right-translocation $\phi(j,i)$, and vice versa. The length of a translocation $\phi(i,j)$ is the number of elements between i and j, including j, that is, $|i-j|$.

We use the term *sequence* to refer to an ordered list. Our is focus is on finite sequences. The length of a sequence s, denoted by $|s|$, is the number of its elements. We write a sequence by separating its elements by commas and enclosing them inside a pair of parentheses. Furthermore, the ith element of a sequence s is denoted by s_i. For example, for a sequence s, we write $s = (s_1,\ldots,s_{|s|})$. Nevertheless, for some types of sequences, we may use a different notation that is particular for that type. For instance, as explained above, a permutation π of $[n]$ is typically written as $\pi = (\pi(1),\ldots,\pi(n))$.

A sequence $\tau = (\tau_1,\ldots,\tau_{|\tau|})$ of permutations is a *transform*. The sequence τ is a *transform converting* π *to* σ if $\pi\tau_1\cdots\tau_{|\tau|} = \sigma$. Note that the permutations τ_i are multiplied on the right. On occasion, we may deal with transforms whose elements are multiplied on the left. Whenever this is the case, it will be explicitly pointed out in the text. By a *transform of* π, we mean a transform that converts π to e. A *transposition transform* is a transform whose elements are transpositions. *Translocation transforms* and *adjacent-transposition transforms* are defined similarly. If the type of the transform is clear from the context, it will not be described in more detail.

1.3.2 Definitions Regarding Rankings

A ranking is a list of candidates arranged in order of preference (or according to some other criterion) with the first candidate being the most preferred and the last candidate being the least preferred.

Consider the set of all possible rankings of n candidates. Via an arbitrary, but fixed, injective mapping from the set of candidates to $\{1,\ldots,n\}$, each ranking may be represented as a permutation. The mapping is often implicit and

we usually equate rankings of n candidates with permutations in $\mathbb{S}_n$. This is equivalent to assuming that the set of candidates is the set $[n]$.

For a ranking $\pi \in \mathbb{S}_n$ and $a, b \in [n]$, π is said to *rank a before b* or *higher than b* if $\pi^{-1}(a) < \pi^{-1}(b)$. For brevity, we may show this relationship as $a <_\pi b$. Two rankings π and σ *agree* on the relative order of a pair $\{a, b\}$ of elements if both rank a before b or both rank b before a. Furthermore, the two rankings π and σ *disagree* on the relative order of a pair $\{a, b\}$ if one ranks a before b and the other ranks b before a. For example, consider $\pi = (1, 2, 3, 4)$ and $\sigma = (4, 2, 1, 3)$. We have that $4 <_\sigma 1$ and that π and σ agree on $\{2, 3\}$ but disagree on $\{1, 2\}$.

1.3.3 Definitions Regarding Graphs

We assume that the reader is familiar with basic concepts and definitions in graph theory. Nevertheless, for completeness, we state some simple facts and definitions.

The vertex set of a graph G is denoted by $V(G)$, and its edge set is denoted by $E(G)$. An edge with endpoints u and v is denoted by (uv). The subgraph of G induced by the vertices in a set $S \subseteq V(G)$ is denoted by $G[S]$. The degree of a vertex v in G is denoted by $\deg_G(v)$ or, if there is no ambiguity, by $\deg(v)$. Deletion of an edge t from a graph G is denoted by $G - t$ and deletion of a vertex v and its incident edges from G is denoted by $G - v$. The same notions can be defined for multigraphs – graphs in which there may exist multiple edges with the same endpoints.

We say that an edge t in G is a *cut edge* for two vertices a and b, denoted by a,b-cut edge, if in $G - t$ there exists no path from a to b. The well-known Menger's theorem [41] asserts that the minimum number of edges one needs to delete from G to disconnect a from b is also the maximum number of pairwise edge-disjoint paths between a and b. This theorem holds for multigraphs as well.

An *embedding* is a drawing of a graph in the plane such that no two edges cross. An embedding of $\mathrm{dig}(\pi)$ can be obtained by placing vertices of disjoint cycles on disjoint circles as seen in Figure 1.1. This embedding is also referred to as $\mathrm{dig}(\pi)$. If the direction of the edges of $\mathrm{dig}(\pi)$ are not explicitly indicated, we assume a clockwise direction and treat $\mathrm{dig}(\pi)$ as an undirected graph.

1.3.4 Distances on Permutations

A *metric* on a set S is a function $\mathsf{d} : S \times S \to \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ denotes the set of nonnegative reals, such that for all $x, y, z \in S$,

1. $\mathsf{d}(x, y) = 0$ if and only if $x = y$,
2. $\mathsf{d}(x, y) = \mathsf{d}(y, x)$, and
3. $\mathsf{d}(x, y) \leq \mathsf{d}(x, z) + \mathsf{d}(z, y)$.

If the first condition is not satisfied but the other two are, then d is a pseudo-metric. A *distance function* or simply a *distance* is a metric or a pseudo-metric. Our discussion focuses on distances on $\mathbb{S}_n$. Below, we review some of the most commonly known such distances. The reader is referred to [38] and [42] for more details.

In addition to the properties stated above, it is often required for a distance on $\mathbb{S}_n$ to have some additional properties. One of the most useful is the notion of *invariance*: A distance d on $\mathbb{S}_n$ is *right-invariant* if, for all $\pi, \sigma, \omega \in \mathbb{S}_n$, we have $\mathsf{d}(\pi, \sigma) = \mathsf{d}(\pi\omega, \sigma\omega)$. Similarly, d is *left-invariant* if $\mathsf{d}(\pi, \sigma) = \mathsf{d}(\omega\pi, \omega\sigma)$. A distance that is both left-invariant and right-invariant is a *bi-invariant* distance. Intuitively, a right-invariant distance is invariant with respect to *reordering* of elements and a left-invariant distance is invariant with respect to *relabeling* of elements.

Some distances can be described in terms of transforms that convert one permutation to another. We use the following definitions when defining such distances as well as elsewhere throughout this work. We use $\mathbb{T}(\pi, \sigma)$ to denote the set of transposition transforms that convert π to σ. That is, $\mathbb{T}(\pi, \sigma)$ is the set of transforms $\tau = (\tau_1, \ldots, \tau_{|\tau|})$ such that $\tau_i \in \mathbb{T}_n$ and $\pi\tau_1 \cdots \tau_{|\tau|} = \sigma$. The set of adjacent-transposition transforms that convert π to σ is denoted by $\mathbb{A}(\pi, \sigma)$.

1.3.4.1 Kendall τ Distance

The Kendall τ distance between two permutations π and σ is denoted by $\mathsf{d}_K(\pi, \sigma)$ and is defined as the minimum number of swaps of adjacent elements required to transform π into σ. In other words,

$$\mathsf{d}_K(\pi, \sigma) = \min\{|\tau| : \tau \in \mathbb{A}(\pi, \sigma)\}.$$

An *inversion* of π is a pair $\{a, b\}$ such that $a < b$ but $\pi^{-1}(b) < \pi^{-1}(a)$. For example, $\{2, 4\}$ is an inversion of $(4, 1, 2, 3)$. An *odd permutation* is a permutation with an odd number of inversions. Similarly, an *even permutation* is a permutation with an even number of inversions.

A *relative inversion* of π and σ is a pair $\{a, b\}$ such that $\pi^{-1}(a) < \pi^{-1}(b)$ but $\sigma^{-1}(b) < \sigma^{-1}(a)$. The set of all such pairs is denoted by $I(\pi, \sigma)$:

$$I(\pi, \sigma) = \left\{\{a, b\} : \pi^{-1}(a) < \pi^{-1}(b), \sigma^{-1}(b) < \sigma^{-1}(a)\right\}.$$

If π and σ are viewed as rankings, a relative inversion is pair on which π and σ disagree. The following lemma, which is well-known, shows that the Kendall τ distance can be interpreted as the number of disagreements between two rankings.

Lemma 1.1. *For $\pi, \sigma \in \mathbb{S}_n$,*

$$\mathsf{d}_K(\pi, \sigma) = |I(\pi, \sigma)|.$$

Proof. Consider a sequence $\tau = (\tau_1, \ldots, \tau_k) \in \mathbb{A}(\pi, \sigma)$, with $k = \mathsf{d}_K(\pi, \sigma)$. Let $\pi_j = \pi\tau_1 \cdots \tau_j$. Each τ_i decreases the number of inversions by at most one. Hence,

$$|I(\pi_j, \sigma)| \geq |I(\pi_{j-1}, \sigma)| - 1$$

and thus

$$0 = |I(\pi_k, \sigma)| \geq |I(\pi, \sigma)| - k.$$

Since $k = \mathsf{d}_K(\pi, \sigma)$, we obtain

$$\mathsf{d}_K(\pi, \sigma) \geq |I(\pi, \sigma)|.$$

On the other hand, it is easy to see that one can find $\tau_i, i \in [k]$, in such a way that each τ_i decreases the number of inversions by one. For example, Bubble Sort [43] is one such well-known algorithm. Hence,

$$\mathsf{d}_K(\pi, \sigma) = |I(\pi, \sigma)|.$$

□

It easily follows from the definition of the Kendall τ distance that it is a left-invariant distance.

1.3.4.2 Cayley Distance

The *Cayley distance* between two permutations, $\mathsf{d}_T(\pi, \sigma)$, is the number of swaps required to convert one permutation to the other. In other words, it is the minimum length of a transposition transform converting π to σ,

$$\mathsf{d}_T(\pi, \sigma) = \min\{|\tau| : \tau \in \mathbb{T}(\pi, \sigma)\}.$$

Note that unlike for the case of the Kendall τ distance, here, one is allowed to use all transpositions and not only adjacent transpositions. The Cayley distance is also known as the *transposition distance.*

From the definition of the Cayley distance it follows that it is left-invariant. However it is also right-invariant, and therefore, it is bi-invariant. The right-invariance of the Cayley distance can be proved using the fact that for every permutation ω and transposition $\langle a\ b\rangle$, we have $\omega\langle a\ b\rangle = \langle\omega(a)\ \omega(b)\rangle\omega$.

A transposition $\langle a\ b\rangle$ applied to a permutation $\pi \in \mathbb{S}_n$ increases (resp. decreases) the number of cycles of π by one if a and b are in the same cycle (resp. different cycles) of π. The distance $\mathsf{d}_T(\pi, e)$ equals n minus the number of cycles

of π as each transposition can increase the number of cycles by one. This result is due to Cayley [40]. From the left-invariance of the Cayley distance, it follows that $\mathsf{d}_T(\pi,\sigma) = \mathsf{d}_T(\sigma^{-1}\pi, e)$. Thus the distance $\mathsf{d}_T(\pi,\sigma)$ equals n minus the number of cycles of $\sigma^{-1}\pi$.

Finally, we note that each transposition applied to a permutation changes the number of inversions by an odd value. Hence, the Cayley distance of an odd (resp. even) permutation to the identity is odd (resp. even).

1.3.4.3 Hamming Distance

The Hamming distance between permutations π and σ, denoted by $\mathsf{d}_H(\sigma,\pi)$, is the number of positions i at which $\pi(i)$ and $\sigma(i)$ differ. The Hamming distance is also bi-invariant.

Suppose that $\{c_1,\ldots,c_k\}$ is the set of the cycles of π with length more than 1. The Hamming distance $\mathsf{d}_H(\pi,e)$ equals $\sum_{i=1}^{k}|c_i|$. From §1.3.4.2, we have $\mathsf{d}_T(\pi,\sigma) = \sum_{i=1}^{k}(|c_i|-1)$. Hence,

$$\mathsf{d}_H(\pi,e) = \mathsf{d}_T(\pi,e) + C_2(\pi),$$

where $C_2(\pi)$ is the number of cycles of π with length at least 2. From the left-invariance of d_H and d_T, we find

$$\mathsf{d}_H(\pi,\sigma) = \mathsf{d}_T(\pi,\sigma) + C_2(\sigma^{-1}\pi). \tag{1.1}$$

When transforming π to σ using transpositions, each transposition decreases the Hamming distance between the two permutations by at most two. Hence, $\mathsf{d}_T(\pi,\sigma) \geq \mathsf{d}_H(\pi,\sigma)/2$. Sorting a permutation of length d requires at most d transpositions. Thus, $\mathsf{d}_T(\pi,\sigma) \leq \mathsf{d}_H(\pi,\sigma)$. These inequalities result in

$$\frac{1}{2}\mathsf{d}_H(\pi,\sigma) \leq \mathsf{d}_T(\pi,\sigma) \leq \mathsf{d}_H(\pi,\sigma). \tag{1.2}$$

In fact, if $\mathsf{d}_H(\pi,\sigma) > 0$, then $\mathsf{d}_T(\pi,\sigma) \leq \mathsf{d}_H(\pi,\sigma) - 1$. The inequalities given in (1.2) can also be obtained by showing that $0 \leq C_2(\sigma^{-1}\pi) \leq \mathsf{d}_T(\pi,\sigma)$.

1.3.4.4 Spearman's Footrule and Rank Correlation

Spearman's Footrule, d_F, and *Spearman's rank correlation*, d_R, are defined as

$$\mathsf{d}_F(\pi,\sigma) = \sum_{i=1}^{n}\left|\pi^{-1}(i) - \sigma^{-1}(i)\right|, \tag{1.3}$$

$$\mathsf{d}_R(\pi,\sigma) = \sqrt{\sum_{i=1}^{n}\left(\pi^{-1}(i) - \sigma^{-1}(i)\right)^2}. \tag{1.4}$$

Both distances are bi-invariant [42].

More generally, one can define the l_p distance on permutations as

$$\left(\sum_{i=1}^{n} \left|\pi^{-1}(i) - \sigma^{-1}(i)\right|^p\right)^{1/p}.$$

Diaconis and Graham [23] proved the following inequalities:

$$\mathsf{d}_K(\pi,\sigma) + \mathsf{d}_T(\pi,\sigma) \le \mathsf{d}_F(\pi,\sigma) \le 2\mathsf{d}_K(\pi,\sigma).$$

1.3.4.5 Ulam and Levenshtein Distances

The *Ulam distance* between two permutations π and σ of $[n]$, denoted by $\mathsf{d}_U(\pi,\sigma)$, equals $n-\mathrm{lcs}(\pi,\sigma)$, where $\mathrm{lcs}(\pi,\sigma)$ is the length of the longest common subsequence[1] of π and σ.

The length of the longest common subsequence of two permutations is left-invariant. To prove this claim, let us consider three arbitrary permutations $\sigma,\pi,\omega \in \mathbb{S}_n$ with $\mathrm{lcs}(\sigma,\pi) = k$. There exists a longest common subsequence $i_1, i_2, \ldots, i_k$ of σ and π. The subsequence $\omega(i_1),\omega(i_2),\ldots,\omega(i_m)$ is a subsequence of both $\omega\sigma$ and $\omega\pi$, and thus $\mathrm{lcs}(\omega\sigma,\omega\pi) \ge \mathrm{lcs}(\sigma,\pi)$. On the other hand, by considering the permutations $\omega\sigma,\omega\pi$, and ω^{-1} instead of σ,π, and ω, it can also be shown that $\mathrm{lcs}(\sigma,\pi) \ge \mathrm{lcs}(\omega\sigma,\omega\pi)$. This proves that $\mathrm{lcs}(\cdot,\cdot)$ is left-invariant. Hence, the Ulam distance is left-invariant.

The following proposition relates the Ulam distance to translocation transforms.

Proposition 1.2. *For $\pi,\sigma \in \mathbb{S}_n$, the distance $\mathsf{d}_U(\pi,\sigma)$ equals the minimum length of a translocation transform converting π to σ.*

Proof. The minimum length of a translocation transform converting π to σ equals the minimum length of a translocation transform converting $\sigma^{-1}\pi$ to e. Additionally, $\mathsf{d}_U(\pi,\sigma) = \mathsf{d}_U(\sigma^{-1}\pi, e)$. Hence, it suffices to prove that $\mathsf{d}_U(\omega,e)$ equals the minimum length of a translocation transform converting ω to e, where $\omega = \sigma^{-1}\pi$.

Let $\mathrm{lcs}(\omega) = \mathrm{lcs}(\omega,e)$. Note that $\mathrm{lcs}(\omega)$ equals the length of a longest increasing subsequence of ω. Let S_l denote the elements of one such subsequence. Furthermore, let d denote the minimum length of a translocation transform converting ω to e. It is possible to transform ω into e with $n-\mathrm{lcs}(\omega)$ translocations by moving the elements of the set $[n]\backslash S_l$. Hence, $d \le \mathsf{d}_U(\omega,e)$.

Next, we show that $d \ge \mathsf{d}_U(\omega,e) = n-\mathrm{lcs}(\omega)$. We start with ω and transform it to e by applying a sequence of translocations of length d. Every translocation increases the length of the longest increasing subsequence by at most one. Hence, we need at least $n-\mathrm{lcs}(\omega)$ translocations to transform ω into e and thus $d \ge \mathsf{d}_U(\omega,e)$. □

[1] Here, as anywhere else in this work, we assume that one may choose, according to some arbitrary but fixed rule, *one* longest common subsequence if the longest common sequence is not unique.

The Ulam distance is closely related to *Levenshtein's insertion/deletion distance*, denoted by $\rho(u,v)$, and defined as the number of deletions and insertions required to transform u into v. Levenshtein [44] showed that, for sequences of length n, $\rho(u,v) = 2(n - \mathrm{lcs}(u,v))$. This equality also holds for permutations in $\mathbb{S}_n$ and thus

$$\rho(\sigma,\pi) = 2\mathsf{d}_U(\sigma,\pi).$$

This result may be also deduced directly, by observing that a translocation consists of a deletion and an insertion.

We study next the relationships between the Ulam distance and the other distances mentioned above. These relationships will be useful for our analysis in Chapter 5 in the context of designing error-correcting codes for flash memories.

A translocation of length ℓ can be represented as ℓ adjacent transpositions. We have $\ell \le n-1$. Thus, $\mathsf{d}_K(\pi,\sigma) \le (n-1)\mathsf{d}_U(\pi,\sigma)$. Furthermore, each adjacent transposition is a translocation. Hence,

$$\frac{1}{n-1}\mathsf{d}_K(\sigma,\pi) \le \mathsf{d}_U(\sigma,\pi) \le \mathsf{d}_K(\sigma,\pi). \tag{1.5}$$

Both the upper bound and the lower bound are tight: the upper bound is achieved for π obtained from σ via a single adjacent transposition, while the lower bound is achieved for, say, $\sigma = e$ and $\pi = (2,3,\ldots,n,1)$. It is also straightforward to show that the diameter of $\mathbb{S}_n$ with respect to the Ulam distance equals $n-1$.

A similar pair of bounds may be shown to hold for the Ulam distance and the Hamming distance between two permutations.

Let $F(\pi,\sigma) = \{i \in [n] : \pi(i) = \sigma(i)\}$. The subsequence of σ consisting of elements $\sigma(i), i \in F(\sigma,\pi)$, is also a subsequence of π and thus $\mathsf{d}_U(\sigma,\pi) = n - \mathrm{lcs}(\sigma,\pi) \le n - |F(\sigma,\pi)| = \mathsf{d}_H(\sigma,\pi)$. Furthermore, since for any two permutations $\pi,\sigma \in \mathbb{S}_n$ one has $\mathsf{d}_H(\pi,\sigma) \le n$, it follows that $\mathsf{d}_H(\pi,\sigma) \le n\mathsf{d}_U(\pi,\sigma)$. Thus,

$$\frac{1}{n}\mathsf{d}_H(\pi,\sigma) \le \mathsf{d}_U(\pi,\sigma) \le \mathsf{d}_H(\pi,\sigma). \tag{1.6}$$

These inequalities are sharp. For the upper-bound, consider $\pi = (1,2,\ldots,n)$ and $\sigma = (n,\ldots,2,1)$, with n odd. For the lower-bound, let $\pi = (1,2,\ldots,n)$ and $\sigma = (2,3,\ldots,n,1)$ so that $\mathsf{d}_H(\pi,\sigma) = n$ and $\mathsf{d}_U(\pi,\sigma) = 1$.

Finally, we consider the relationship between the Ulam distance and the Cayley distance. Note that each transposition may be viewed as two translocations, implying that $\mathsf{d}_U(\pi,\sigma) \le 2\mathsf{d}_T(\sigma,\pi)$. It is also immediate that $\frac{1}{n-1}\mathsf{d}_T(\pi,\sigma) \le \mathsf{d}_U(\pi,\sigma)$. Hence, we have

$$\frac{1}{n-1}\mathsf{d}_T(\pi,\sigma) \le \mathsf{d}_U(\pi,\sigma) \le 2\mathsf{d}_T(\sigma,\pi).$$

Chapter 2

Weighted Kendall Distance

2.1 Introduction

Rank aggregation, sometimes referred to as *ordinal data fusion*, is a classical problem frequently encountered in the social sciences, web search and Internet service studies, expert opinion analysis, and economics [15, 28, 45–48].

The problem can be succinctly described as follows: a set of "voters" or "experts" is presented with a set of candidates (objects, individuals, movies, etc.). Each voter's task is to produce a ranking, that is, an arrangement of the candidates in which the candidates are ranked from the most preferred to the least preferred. The voters' rankings are then passed to an aggregator. The aggregator outputs a single ranking, the *aggregate ranking*, to be used as a representative of all votes.

If there are only two candidates, a natural solution exists: the candidate that is preferred by a majority of voters is ranked first and the other candidate is ranked second. The situation becomes significantly more complex when three or more candidates are considered. Two of the most obvious extensions of vote aggregation for two candidates to the case of more than two candidates are the plurality rule and the Condorcet method (pairwise majority count). In the first case, candidates are ranked based on the number of times they appear at the top of the a voter's ranking. In the second case, among two candidates a and b, candidate a is preferred to candidate b if a is preferred to b by a majority of voters. Unfortunately, both methods are plagued by a number of problems that have cast doubt on the plausibility of fair vote aggregation. Examples include the famous Condorcet paradox [49] which arises when majority preference is not transitive (i.e., for example, a is preferred to b, b to c, and c to a).

To mitigate such problems, two important categories of rank aggregation methods were studied in the past. These are *score-based* methods and *distance-based* methods. In score-based methods, the first variant of which was proposed by Borda [50], each candidate is assigned a score based on its position in each of the votes (rankings). The candidates are then ranked based on their total score. One argument in support of using Borda's count method is that it ranks highly those candidates supported at least to a certain extent by almost all voters, rather than candidates who are ranked highly only by the simple majority

of voters. In distance-based methods [15], the aggregate is deemed to be the ranking at the smallest cumulative distance from the votes. This approach can be thought of as finding the median of the set of voters' rankings. Well-known distance measures for rank aggregation include the Kendall τ, the Cayley distance, and Spearman's Footrule [38].

The most important aspect of distance-based rank aggregation is to choose an appropriate distance function. To address this issue, Kemeny [15, 16] presented a set of intuitively justifiable axioms that a distance measure must satisfy to be deemed suitable for aggregation purposes, and showed that only one distance measure satisfies the axioms – namely, the Kendall τ distance. Recall that the Kendall τ distance between two rankings is the smallest number of swaps of adjacent elements that transforms one ranking into the other.

Unfortunately, the Kendall τ is not flexible enough to take into account two important factors. First, in many applications different positions in a ranking are of different degrees of importance. For example, the top of a ranking may be more important than the bottom and so changes to the top of the ranking must induce a larger distance. Second, when transposing elements of the ranking, transposing elements that are similar must induce a smaller distance than those that are dissimilar. Devising a distance function that can reflect the first factor is the subject of this chapter. In Chapter 3, we propose a distance that takes into account the second factor and discuss its applications.

In the next subsection, we discuss in more details the relative importance of different positions in a ranking.

2.1.1 Motivation – Top vs. Bottom

Consider the ranking π of the "World's 10 best cities to live in," according to a report composed by the Economist Intelligence Unit [51]:

$$\pi = (\text{Melbourne, Vienna, Vancouver, Toronto, Calgary,}$$
$$\text{Adelaide, Sydney, Helsinki, Perth, Auckland}).$$

Now consider two other rankings that both differ from π by one swap of adjacent entries:

$$\pi' = (\text{Melbourne, Vienna, Vancouver, Calgary, Toronto,}$$
$$\text{Adelaide, Sydney, Helsinki, Perth, Auckland}),$$
$$\pi'' = (\text{Vienna, Melbourne, Vancouver, Toronto, Calgary,}$$
$$\text{Adelaide, Sydney, Helsinki, Perth, Auckland}).$$

The astute reader probably immediately noticed that the top candidate was changed in π'', but otherwise took some time to realize where the adjacent swap appeared in π'. This is a consequence of the well-known fact that humans pay

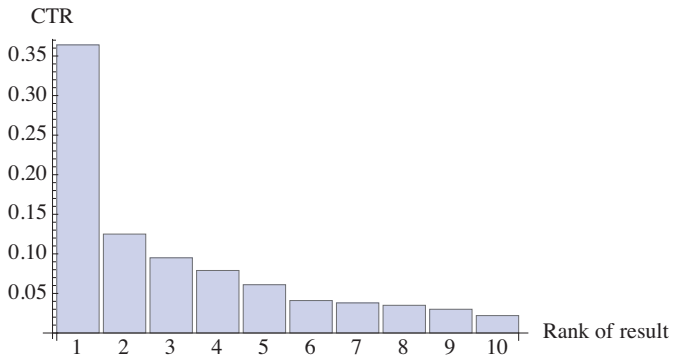


Figure 2.1: Click-through rates of webpages appearing on the first page of Google search.

more attention to the top of the list than any other location in the ranking, and hence notice changes in higher positions easier.[1] Note that the Kendall τ distance between π and π' and between π and π'' is one, but it would appear reasonable to require that the distance between π and π'' be larger than that between π and π', as the corresponding swap occurred in a *more significant* (higher ranked) position in the list.

As a second example, consider the well-studied notion of *Click-through rates* (CTRs) of webpages in search engine results pages (SERPs). The CTR of a link is the number of clicks on that link divided by the number of times that it is displayed [52]. A recent study by Optify Inc. [30] showed that the difference between the average CTR of the first (highest-ranked) result and the average CTR of the second (runner-up) result is very large, and much larger than the corresponding difference between the average CTRs of the lower ranked items (see Figure 2.1). Hence, in terms of directing search engine traffic, swapping higher-ranked adjacent pairs of search results has a larger effect on the performance of Internet services than swapping lower-ranked search results.

The aforementioned findings should be considered when forming an aggregate ranking of webpages. For example, in studies of CTRs, one is often faced with questions regarding traffic flow from search engines to webpages. One may think of a set of keywords, each producing a different ranking of possible webpages, with the aggregate representing the median. Based on Figure 2.1, if a webpage is ranked in the bottom half, its exact position is not as relevant as when it is ranked in the top half. Furthermore, a webpage appearing roughly half of the time at the top and roughly half of the time at the bottom will generate more incoming traffic than a webpage with persistent average ranking.

We refer to the issue of different importance of different positions in rankings

[1] Note that one may argue that people are equally drawn to explore the highest and lowest ranked items in a list. For example, if about a hundred cities were ranked, it would be reasonable to assume that readers would be more interested in knowing the ten highest ranked and the ten lowest ranked cities, rather than the cities occupying positions 41 to 60. These positional differences may also be addressed within the framework proposed in this chapter.

as the "top-vs-bottom" problem. Besides the importance in emphasizing the relevance of the top of the list, distance measures that penalize perturbations at the top of the list more than perturbations at the bottom of the list have another important application in practice – eliminating negative outliers. As will be shown in subsequent sections, top-vs-bottom distance measures allow candidates to be highly ranked in the aggregate even though they have a certain (small) number of highly negative rankings. The policy of eliminating outliers before rating items or individuals is a well-known one, but has not been considered in the social choice literature in the context of distance-based rank aggregation.

To address the top-vs-bottom, we axiomatically describe a class of distance functions by assigning different weights to different adjacent transpositions, termed the weighted Kendall distance. Furthermore, we show that the proposed distance functions can be computed in polynomial time in two important special cases and provide a polynomial-time 2-approximation algorithm for the general case.

The remainder of the chapter is organized as follows. Related work is reviewed in §2.2. In §2.3, we present Kemeny's axioms and show that for rankings with no ties one of Kemeny's axioms in redundant. The weighted Kendall distance, as well as its axiomatic definitions, is introduced in §2.4. We devote §2.5 to the computational aspects of the weighted Kendall distance. Finally, in §2.6, we discuss how the weighted Kendall distance can be used for rank aggregation.

2.2 Related Work

The need for distance measures that do not treat different positions in rankings similarly, as well as the inadequacy of the Kendall τ distance to be used for solving such problems, is well known, see, for example [24–29]. In this section, we review some of the measures of correlation/discordance between rankings that were proposed in the literature, with the goal of addressing the aforementioned problem.

Shieh [25] proposes a function SH_w on rankings defined as

$$SH_w(\pi,\sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} \mathcal{I}\big(\sigma^{-1}(\pi(j)) < \sigma^{-1}(\pi(i))\big)$$

where w_{ij} is the weight assigned to the pair $\{i,j\}$ with $w_{ij} = w_{ji}$ and $w_{ii} = 0$, and where $\mathcal{I}(\text{Condition})$ equals 1 if Condition is true and equals 0 otherwise. If the weights w_{ij} for small i and j are chosen to be larger than the weights for large i and j, then SH_w penalizes relative inversions that involve elements that are ranked high by π more than elements that are ranked low.

An important drawback of SH_w is that it is not symmetric and thus not a

distance. For example,

$$SH_w((a,b,c),(b,c,a)) = w_{12} + w_{13},$$
$$SH_w((b,c,a),(a,b,c)) = w_{13} + w_{23}.$$

To address this problem, we can symmetrize SH_w by defining a function $\overline{SH}_w$ as

$$\overline{SH}_w(\pi,\sigma) = \frac{SH_w(\pi,\sigma) + SH_w(\sigma,\pi)}{2}.$$

However, it is not clear what properties such a function has and if it is in fact a metric for all nonnegative weights w.

Yilmaz et al. [26] propose a measure of discordance between rankings that can be expressed as the expected outcome of an experiment, which is presented below with slight modification. Given two rankings π and σ,

1. Pick an element a of π, other than its first element, randomly and uniformly.
2. Pick a second element b of π such that it is ranked before a by π, randomly and uniformly.
3. Return 1 if b is ranked after a in σ. Otherwise return 0.

Let the expected outcome of the experiment be denoted by $YA(\pi,\sigma)$. It can be shown that [26]

$$YA(\pi,\sigma) = \frac{1}{n-1}\sum_{j=2}^{n}\frac{C_j(\pi,\sigma)}{j-1},$$

where $C_j(\pi,\sigma)$ is the number of candidates with rank less than j in π that are ranked after $\pi(j)$ in σ.

We show that $YA(\pi,\sigma)$ is, perhaps surprisingly, equal to $SH_w(\pi,\sigma)$ for an appropriate choice of w. Observe that

$$\begin{aligned}
YA(\pi,\sigma) &= \frac{1}{n-1}\sum_{j=2}^{n}\frac{C_j(\pi,\sigma)}{j-1} \\
&= \frac{1}{n-1}\sum_{j=2}^{n}\frac{\left|\left\{x : \pi^{-1}(x) < j, \sigma^{-1}(\pi(j)) < \sigma^{-1}(x)\right\}\right|}{j-1} \\
&= \frac{1}{n-1}\sum_{j=2}^{n}\frac{\left|\left\{i : i < j, \sigma^{-1}(\pi(j)) < \sigma^{-1}(\pi(i))\right\}\right|}{j-1} \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{\mathcal{I}\left(\sigma^{-1}(\pi(j)) < \sigma^{-1}(\pi(i))\right)}{(n-1)(j-1)} \\
&= SH_w(\pi,\sigma)
\end{aligned}$$

with $w_{ij} = \frac{1}{(n-1)(j-1)}$ for $i < j$.

Similar to $SH_w(\pi,\sigma)$, $YA(\pi,\sigma)$ can be symmetrized. However the properties of the symmetrized version of $YA(\pi,\sigma)$ are not studied in [26].

Kumar and Vassilvitskii [28] introduced a distance on rankings based on the Kendall τ distance as follows. For a sequence $w_i, i \in [n-1]$, let

$$f_w(i,j) = \begin{cases} \sum_{r=i}^{j-1} w_r, & \text{if } i < j, \\ \sum_{r=j}^{i-1} w_r, & \text{if } j < i, \\ 0, & \text{if } i = j. \end{cases} \tag{2.1}$$

The distance proposed in [28] can be written as

$$KV_{w,D}(\pi,\sigma) = \sum_{i,j} \frac{f_w(\pi^{-1}(i),\sigma^{-1}(i))}{|\pi^{-1}(i)-\sigma^{-1}(i)|} \frac{f_w(\pi^{-1}(j),\sigma^{-1}(j))}{|\pi^{-1}(j)-\sigma^{-1}(j)|} \mathcal{I}\big(\pi^{-1}(i) < \pi^{-1}(j), \sigma^{-1}(j) < \sigma^{-1}(i)\big).$$

The term $\frac{f_w(\pi^{-1}(i),\sigma^{-1}(i))}{|\pi^{-1}(i)-\sigma^{-1}(i)|}$ can be viewed as the average weight of moving i from its position in π to its position in σ. A similar interpretation holds for the term $\frac{f_w(\pi^{-1}(j),\sigma^{-1}(j))}{|\pi^{-1}(j)-\sigma^{-1}(j)|}$. However, it is not clear why these terms should be chosen to act multiplicatively and what the effect of such a choice is on the properties of the proposed distance.

Sun et al. [53] propose an intuitive distance on rankings which is essentially a weighted version of Spearman's footrule. The distance is[2]

$$SL_w(\pi,\sigma) = \sum_{r=1}^{n} f_w(\pi^{-1}(i),\sigma^{-1}(i)),$$

where f_w is defined in (2.1).

Several other distances and measures of correlation/discordance are also proposed in the literature. For a survey, see [31]. The main difference between our approach and the previously proposed methods and techniques for addressing the top-vs-bottom issue is that our approach has an axiomatic underpinning; the distance that we propose is the unique solution to a set of axioms that has Kemeny's axioms as its basis. Yet the definition of this distance is simple and similar to that of the Kendall τ distance. Furthermore, we show that when used for the purpose of rank aggregation, the proposed distance results in aggregate rankings which take into account the top-vs-bottom issue.

2.3 Reduced Kemeny Axioms

In [15] Kemeny proposed a distance-based rank aggregation approach. Given a distance function d over permutations and a set Σ of votes, the distance-based aggregation problem can be stated as follows: find the ranking π^* that

[2]This distance is in fact a special case of the distance that we introduce in the next chapter. See Lemma 3.15.

minimizes the cumulative distance from Σ, i.e.,

$$\pi^* = \arg\min_{\pi \in \mathbb{S}_n} \sum_{\sigma \in \Sigma} \mathsf{d}(\pi, \sigma). \tag{2.2}$$

In other words, the goal is to find a ranking π that represents the median of the set Σ of votes. The choice of the distance function d is an important aspect of distance-based rank aggregation and the focus of this chapter.

Kemeny presented a set of axioms that a distance function for rank aggregation should satisfy and proved that the only distance that satisfies the axioms is the Kendall τ.

In Kemeny's work, rankings are allowed to have ties, while in this work, our focus is on rankings with no ties. We show that if ties are not allowed, after removing one of Kemeny's axioms, the Kendall τ distance is still the unique distance that satisfies this reduced set of axioms.

A critical concept in Kemeny's axioms is the idea of "betweenness," defined below. Other definitions used in this chapter can be found in §1.3.

Definition 2.1. A ranking ω is *between* two rankings π and σ, denoted by π–ω–σ, if for each pair $\{a, b\}$ of candidates, ω either agrees with π or σ or both. The rankings $\pi_1, \ldots, \pi_s$ are *on a line*, denoted by π_1–π_2–$\cdots$–π_s, if for every i, j, and k for which $1 \le i < j < k \le s$, we have π_i–π_j–π_k.

The reduced set of Kemeny's axioms are:

Axioms I

1. d is a metric.
2. d is left-invariant.
3. For any π, σ, and ω, $\mathsf{d}(\pi, \sigma) = \mathsf{d}(\pi, \omega) + \mathsf{d}(\omega, \sigma)$ if and only if ω is between π and σ.
4. The smallest positive distance is one.

Axiom 2 states that relabeling of objects should not change the distance between permutations. In other words, $\mathsf{d}(\sigma\pi, \sigma\omega) = \mathsf{d}(\pi, \omega)$, for any $\pi, \sigma, \omega \in \mathbb{S}_n$. Axiom 3 may be viewed through a geometric lens: the triangle inequality has to be satisfied with equality for all points that lie on a line between π and σ. Axiom 4 is only used for normalization purposes.

Kemeny's original exposition included a fifth axiom which we state for completeness: If two rankings π and σ agree except for a segment of k elements, the position of the segment does not affect the distance between the rankings. Here, a segment is a set of objects that are ranked consecutively, i.e., a substring of the permutation. As an example, this axiom implies that

$$\mathsf{d}((1,2,3,\underbrace{4,5,6}),(1,2,3,\underbrace{6,5,4})) = \mathsf{d}((1,\underbrace{4,5,6},2,3),(1,\underbrace{6,5,4},2,3)),$$

where the segment is underscored by braces. This axiom clearly enforces a property that *is not desirable* for distances designed to address the top-vs-bottom issue: changing the position of the segment in two permutations does not alter their distance. One may hence believe that removing this axiom (as was done in Axioms I) will lead to distance measures capable of handling the top-vs-bottom problem. But as we show below, for rankings without ties, omitting this axiom does not change the outcome of Kemeny's analysis; in other words, the axiom is redundant. This is a rather surprising fact, and we conjecture that the same is true of rankings with ties.

The main result of this section is Theorem 2.6, stating that the unique distance satisfying Axioms I is the Kendall τ distance. The theorem is proved with the help of Lemmas 2.2, 2.3, 2.4, 2.5, as well as Lemma 1.1.

Lemma 2.2. *For any distance measure* d *that satisfies Axioms I, and for any sequence of permutations* $\pi_1, \pi_2, \ldots, \pi_s$ *such that* $\pi_1 – \pi_2 – \cdots – \pi_s$*, one has*

$$\mathsf{d}(\pi_1, \pi_s) = \sum_{k=1}^{s-1} \mathsf{d}(\pi_k, \pi_{k+1}).$$

Proof. The lemma follows from Axiom I.3 by induction. □

Lemma 2.3. *For any* d *that satisfies Axioms I and for* $i \in [n-1]$*, we have*

$$\mathsf{d}\left(\langle i\ i+1\rangle, e\right) = \mathsf{d}\left(\langle 1\ 2\rangle, e\right).$$

Proof. We first show that $\mathsf{d}\left(\langle 2\ 3\rangle, e\right) = \mathsf{d}\left(\langle 1\ 2\rangle, e\right)$. Repeating the same argument used for proving this special case gives $\mathsf{d}\left(\langle i\ i+1\rangle, e\right) = \mathsf{d}\left(\langle i-1\ i\rangle, e\right) = \cdots = \mathsf{d}\left(\langle 1\ 2\rangle, e\right)$.

To show that $\mathsf{d}\left(\langle 2\ 3\rangle, e\right) = \mathsf{d}\left(\langle 1\ 2\rangle, e\right)$, we evaluate $\mathsf{d}(\pi, e)$ in two ways, where we choose $\pi = (3, 2, 1, 4, 5, \ldots, n)$.

On the one hand, note that $\pi – \omega – \eta – e$, where $\omega = \pi\langle 1\ 2\rangle = (2, 3, 1, 4, 5, \ldots, n)$ and $\eta = \omega\langle 2\ 3\rangle = (2, 1, 3, 4, 5, \ldots, n)$. As a result,

$$\begin{aligned}
\mathsf{d}(\pi, e) &= \mathsf{d}(\pi, \omega) + \mathsf{d}(\omega, \eta) + \mathsf{d}(\eta, e) \\
&= \mathsf{d}(\omega^{-1}\pi, e) + \mathsf{d}(\eta^{-1}\omega, e) + \mathsf{d}(\eta, e) \\
&= \mathsf{d}(\langle 1\ 2\rangle, e) + \mathsf{d}(\langle 2\ 3\rangle, e) + \mathsf{d}(\langle 1\ 2\rangle, e), \qquad (2.3)
\end{aligned}$$

where the first equality follows from Lemma 2.2, while the second is a consequence of the left-invariance property of the distance measure.

On the other hand, note that $\pi – \alpha – \beta – e$, where $\alpha = \pi\langle 2\ 3\rangle = (3, 1, 2, 4, 5, \ldots, n)$ and $\beta = \alpha\langle 1\ 2\rangle = (1, 3, 2, 4, 5, \ldots, n)$. For this case,

$$\begin{aligned}
\mathsf{d}(\pi, e) &= \mathsf{d}(\pi, \alpha) + \mathsf{d}(\alpha, \beta) + \mathsf{d}(\beta, e) \\
&= \mathsf{d}(\alpha^{-1}\pi, e) + \mathsf{d}(\beta^{-1}\alpha, e) + \mathsf{d}(\beta, e) \\
&= \mathsf{d}(\langle 2\ 3\rangle, e) + \mathsf{d}(\langle 1\ 2\rangle, e) + \mathsf{d}(\langle 2\ 3\rangle, e). \qquad (2.4)
\end{aligned}$$

Equations (2.3) and (2.4) imply that $\mathsf{d}(\langle 2\ 3\rangle, e) = \mathsf{d}(\langle 1\ 2\rangle, e)$. □

Lemma 2.4. *For any* d *that satisfies Axioms I,* $\mathsf{d}(\gamma, e)$ *equals the minimum number of adjacent transpositions required to transform* γ *into* e*.*

Proof. For $\pi, \sigma \in \mathbb{S}_n$, let

$$\mathbb{L}(\pi, \sigma) = \{\tau = (\tau_1, \ldots, \tau_{|\tau|}) \in \mathbb{A}(\pi, \sigma) : \pi\text{–}\pi\tau_1\text{–}\pi\tau_1\tau_2\text{–}\cdots\text{–}\sigma\}$$

be the subset of $\mathbb{A}(\pi, \sigma)$ consisting of transposition transforms that convert π to σ by passing through a line. Let s be the minimum number of adjacent transpositions that transform γ into e. Furthermore, let $(\tau_1, \tau_2, \ldots, \tau_s) \in \mathbb{A}(\gamma, e)$ and define $\gamma_i = \gamma\tau_1 \cdots \tau_i, i = 0, \ldots, s$, with $\gamma_0 = \gamma$ and $\gamma_s = e$.

First, we show $\gamma_0\text{–}\gamma_1\text{–}\cdots\text{–}\gamma_s$, that is,

$$(\tau_1, \tau_2, \ldots, \tau_s) \in \mathbb{L}(\gamma, e). \tag{2.5}$$

Suppose this were not the case. Then, there exist $i < j < k$ such that γ_i, γ_j, and γ_k are not on a line, and thus, there exists a pair $\{r, s\}$ for which γ_j disagrees with both γ_i and γ_k. Hence, there exist two transpositions, $\tau_{i'}$ and $\tau_{j'}$, with $i < i' \le j$ and $j < j' \le k$ that swap r and s. We can in this case remove $\tau_{i'}$ and $\tau_{j'}$ from $(\tau_1, \ldots, \tau_s)$ to obtain $(\tau_1, \ldots, \tau_{i'-1}, \tau_{i'+1}, \ldots, \tau_{j'-1}, \tau_{j'+1}, \tau_s) \in \mathbb{A}(\gamma, e)$ with length $s - 2$. This contradicts the optimality of the choice of s. Hence, $(\tau_1, \tau_2, \ldots, \tau_s) \in \mathbb{L}(\gamma, e)$. Then Lemma 2.2 implies that

$$\mathsf{d}(\gamma, e) = \sum_{i=1}^{s} \mathsf{d}(\tau_i, e). \tag{2.6}$$

Lemma 2.3 states that all adjacent transpositions have the same distance from the identity. Since transpositions $\tau_i, 1 \le i \le s$, in (2.6) are adjacent transpositions, $\mathsf{d}(\tau_i, e) = a$ for some $a > 0$ and thus $\mathsf{d}(\gamma, e) = sa$.

In (2.6), the minimum positive distance is obtained when $s = 1$. That is, the minimum positive distance from identity equals a and is obtained when γ is an adjacent transposition. Axiom I.4 states that the minimum positive distance is 1. By left-invariance, this axiom implies that the minimum positive distance of any permutation from the identity is 1. Hence, $a = 1$ and for any $\gamma \in \mathbb{S}_n$,

$$\mathsf{d}(\gamma, e) = \sum_{i=1}^{s} \mathsf{d}(\tau_i, e) = sa = s.$$

□

Lemma 2.5. *For any* d *that satisfies Axioms I, and for* $\pi, \sigma \in \mathbb{S}_n$*, we have*

$$\mathsf{d}(\pi, \sigma) = \min\{s : (\tau_1, \ldots, \tau_s) \in \mathbb{A}(\pi, \sigma)\}.$$

Proof. We have

$$\begin{aligned}\mathsf{d}(\pi,\sigma) &\overset{(a)}{=} \mathsf{d}(\sigma^{-1}\pi, e)\\ &\overset{(b)}{=} \min\left\{s : (\tau_1,\ldots,\tau_s) \in \mathbb{A}(\sigma^{-1}\pi, e)\right\}\\ &\overset{(c)}{=} \min\left\{s : (\tau_1,\ldots,\tau_s) \in \mathbb{A}(\pi,\sigma)\right\},\end{aligned}$$

where (a) follows from the left-invariance of d; (b) follows from Lemma 2.4; and (c) follows from the fact that $(\tau_1,\ldots,\tau_s) \in \mathbb{A}(\pi,\sigma)$ if and only if $(\tau_1,\ldots,\tau_s) \in \mathbb{A}(\sigma^{-1}\pi, e)$. □

Theorem 2.6. *The unique distance* d *that satisfies Axioms I is the Kendall* τ *distance,*

$$\mathsf{d}_K(\pi,\sigma) = \min\left\{s : (\tau_1,\ldots,\tau_s) \in \mathbb{A}(\pi,\sigma)\right\}, \qquad \textit{for } \pi,\sigma \in \mathbb{S}_n.$$

Proof. We show below that d_K satisfies Axiom I.3. Proving that d_K satisfies the other axioms is straightforward. Uniqueness follows from Lemma 2.5.

To show that d_K satisfies Axiom I.3, we use Lemma 1.1 stating that

$$\mathsf{d}_K(\pi,\sigma) = |I(\pi,\sigma)|.$$

Fix $\pi,\sigma \in \mathbb{S}_n$. For any $\omega \in \mathbb{S}_n$, it is clear that

$$I(\pi,\sigma) \subseteq I(\pi,\omega) \cup I(\omega,\sigma). \tag{2.7}$$

Suppose first that ω is not between π and σ. Then there exists a pair $\{a,b\}$ with $a <_\pi b$ and $a <_\sigma b$ but such that $a >_\omega b$. Since $\{a,b\} \notin I(\pi,\sigma)$ but $\{a,b\} \in I(\pi,\omega) \cup I(\omega,\sigma)$, we find that

$$|I(\pi,\sigma)| < |I(\pi,\omega) \cup I(\omega,\sigma)|,$$

and thus

$$\begin{aligned}\mathsf{d}_K(\pi,\sigma) &= |I(\pi,\sigma)|\\ &< |I(\pi,\omega) \cup I(\omega,\sigma)|\\ &\le |I(\pi,\omega)| + |I(\omega,\sigma)|\\ &= \mathsf{d}_K(\pi,\omega) + \mathsf{d}_K(\omega,\sigma).\end{aligned}$$

Hence, if ω is not between π and σ, then

$$\mathsf{d}_K(\pi,\sigma) \neq \mathsf{d}_K(\pi,\omega) + \mathsf{d}_K(\omega,\sigma).$$

Next, suppose ω is between π and σ. This immediately implies that $I(\pi,\omega) \subseteq$

$I(\pi,\sigma)$ and $I(\omega,\sigma) \subseteq I(\pi,\sigma)$. These relations, along with (2.7) imply that

$$I(\pi,\omega) \cup I(\omega,\sigma) = I(\pi,\sigma). \tag{2.8}$$

We claim that $I(\pi,\omega) \cap I(\omega,\sigma) = \varnothing$. To see this, observe that if $\{a,b\} \in I(\pi,\omega) \cap I(\omega,\sigma)$, then the relative rankings of a and b are the same for π and σ and so, $\{a,b\} \notin I(\pi,\sigma)$. The last statement contradicts (2.8) and thus

$$I(\pi,\omega) \cap I(\omega,\sigma) = \varnothing. \tag{2.9}$$

From (2.8) and (2.9), we may write

$$\begin{aligned} \mathsf{d}_K(\pi,\sigma) &= |I(\pi,\sigma)| \\ &= |I(\pi,\omega) \cup I(\omega,\sigma)| \\ &= |I(\pi,\omega)| + |I(\omega,\sigma)| \\ &= \mathsf{d}(\pi,\omega) + \mathsf{d}(\omega,\sigma), \end{aligned}$$

and this completes the proof of the fact that d_K satisfies Axiom I.3. □

A distance d on $\mathbb{S}_n$ is called a *graphic distance* [54] if there exists a graph G with vertex set $\mathbb{S}_n$ such that for $\pi,\sigma \in \mathbb{S}_n$, $\mathsf{d}(\pi,\sigma)$ is equal to the length of the shortest path between π and σ in G. Note that this definition implies that the edge set of G is the set

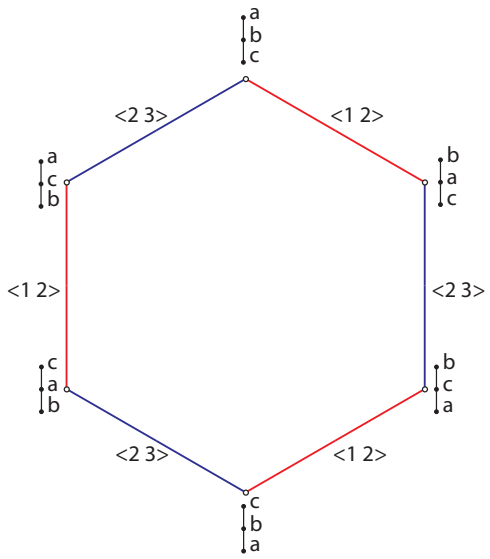
$$\{(\alpha,\beta) : \alpha,\beta \in \mathbb{S}_n, \mathsf{d}(\alpha,\beta) = 1\}.$$

The Kendall τ distance is a graphic distance. To see the validity of this claim, take the corresponding graph to have vertices indexed by permutations, with an edge between each pair of permutations that differ by only one adjacent transposition.

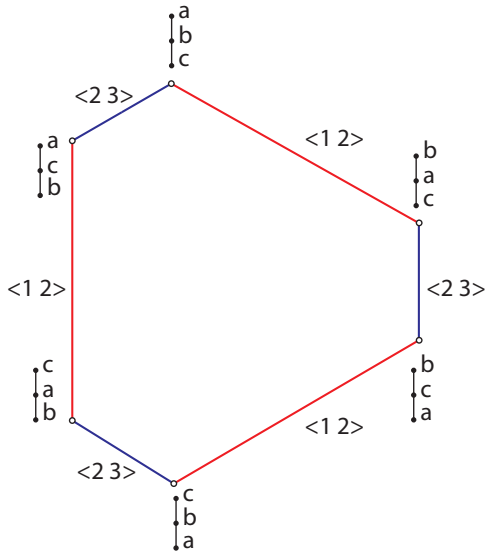
In the next section, we introduce the weighted Kendall distance which may be viewed as the shortest path between permutations over a *weighted graph* (see Figure 2.2 for an illustration), and show how this distance arises from modifying Kemeny's axioms.

2.4 The Weighted Kendall Distance

The proof of the uniqueness of the Kendall τ distance under Axioms I reveals an important insight: the Kendall τ distance arises due to the fact that adjacent transpositions have uniform weights, which is a consequence of the betweenness property used in one of the axioms. To address the top-vs-bottom problem, one must change the uniformity of weights of adjacent transpositions. As we show below, a way to achieve this goal is to redefine the axioms in terms of the



(a)



(b)

Figure 2.2: The graphs for the Kendall τ distance (a), and weighted Kendall distance (b).

betweenness property.

Axioms II

1. d is a pseudo-metric.
2. d is left-invariant.
3. For any π, σ disagreeing on more than one pair of elements, there exists *some* ω, distinct from π and σ and between them, such that $\mathsf{d}(\pi, \sigma) = \mathsf{d}(\pi, \omega) + \mathsf{d}(\omega, \sigma)$.

Axiom II.1 allows for the option that some transpositions do not contribute to the distance between permutations. Intuitively, Axiom II.3 states that there exists at least one point on some line between π and σ, for which the triangle inequality is an equality. In other words, there exists one "shortest line" between two permutations, and not all straight lines are required to be of the same length.

Lemma 2.7. *For any distance* d *that satisfies Axioms II, and for distinct* π *and* σ*, we have*

$$\mathsf{d}(\pi, \sigma) = \min_{\tau \in \mathbb{A}(\pi, \sigma)} \sum_{i=1}^{|\tau|} \mathsf{d}(\tau_i, e).$$

Proof. From the triangle inequality and the left-invariance property of d, we have

$$\mathsf{d}(\pi, \sigma) \leq \min_{\tau \in \mathbb{A}(\pi, \sigma)} \sum_{i=1}^{|\tau|} \mathsf{d}(\tau_i, e).$$

To prove

$$\mathsf{d}(\pi, \sigma) \geq \min_{\tau \in \mathbb{A}(\pi, \sigma)} \sum_{i=1}^{|\tau|} \mathsf{d}(\tau_i, e),$$

we use induction on $\mathsf{d}_K(\pi, \sigma)$, the Kendall τ distance between π and σ.

First, suppose that $\mathsf{d}_K(\pi, \sigma) = 1$, i.e., π and σ disagree on one pair of adjacent elements. Then, we have $\sigma = \pi \langle a \ a+1 \rangle$ for some $a \in [n-1]$. Thus,

$$\mathsf{d}(\pi, \sigma) = \mathsf{d}(\langle a \ a+1 \rangle, e) \geq \min_{\tau \in \mathbb{A}(\pi, \sigma)} \sum_{i=1}^{|\tau|} \mathsf{d}(\tau_i, e),$$

where the equality follows from the left-invariance property of d and the inequality follows from the fact that $(\langle a \ a+1 \rangle) \in \mathbb{A}(\pi, \sigma)$.

Next, suppose that $\mathsf{d}_K(\pi, \sigma) > 1$, i.e., π and σ disagree on more than one pair of adjacent elements, and that for all $\mu, \eta \in \mathbb{S}_n$ with $\mathsf{d}_K(\mu, \eta) < \mathsf{d}_K(\pi, \sigma)$, the lemma holds. Then, there exists ω, distinct from π and σ and between them, such that

$$\begin{aligned} \mathsf{d}(\pi, \sigma) &= \mathsf{d}(\pi, \omega) + \mathsf{d}(\omega, \sigma), \\ \mathsf{d}_K(\pi, \omega) &< \mathsf{d}_K(\pi, \sigma), \\ \mathsf{d}_K(\omega, \sigma) &< \mathsf{d}_K(\pi, \sigma). \end{aligned}$$

By the induction hypothesis, there exist $(\nu_1, \ldots, \nu_k) \in \mathbb{A}(\pi, \omega)$ and $(\nu_{k+1}, \ldots, \nu_s) \in \mathbb{A}(\omega, \sigma)$, for some s and k, such that

$$\mathsf{d}(\pi, \omega) = \sum_{i=1}^{k} \mathsf{d}(\nu_i, e),$$
$$\mathsf{d}(\omega, \sigma) = \sum_{i=k+1}^{s} \mathsf{d}(\nu_i, e),$$

and thus

$$\mathsf{d}(\pi, \sigma) = \sum_{i=1}^{s} \mathsf{d}(\nu_i, e) \geq \min_{\tau \in \mathbb{A}(\pi, \sigma)} \sum_{i=1}^{|\tau|} \mathsf{d}(\tau_i, e),$$

where the inequality follows from the fact that $(\nu_1, \ldots, \nu_s) \in \mathbb{A}(\pi, \sigma)$. □

Definition 2.8. A distance d_φ is a *weighted Kendall distance* if there exists a *weight function* $\varphi : \mathbb{A}_n \to \mathbb{R}_{\geq 0}$ such that

$$\mathsf{d}_\varphi(\pi, \sigma) = \min_{\tau \in \mathbb{A}(\pi, \sigma)} \sum_{i=1}^{|\tau|} \varphi_{\tau_i},$$

where φ_{τ_i} is the weight assigned to transposition τ_i by φ.

The *weight of a transform* τ is denoted by $\mathrm{wt}(\tau)$ and is defined as

$$\mathrm{wt}(\tau) = \sum_{i=1}^{|\tau|} \varphi_{\tau_i}.$$

Hence, $\mathsf{d}_\varphi(\pi, \sigma)$ may be written as

$$\mathsf{d}_\varphi(\pi, \sigma) = \min_{\tau \in \mathbb{A}(\pi, \sigma)} \mathrm{wt}(\tau).$$

Note that a weighted Kendall distance is completely determined by its weight function φ.

Theorem 2.9. *A distance* d *satisfies Axioms II if and only if it is a weighted Kendall distance.*

Proof. It follows immediately from Lemma 2.7 that a distance d satisfying Axioms II is a weighted Kendall distance by letting

$$\varphi_\theta = \mathsf{d}(\theta, e)$$

for every transposition θ taken from the set of adjacent transpositions $\mathbb{A}_n$ in $\mathbb{S}_n$.

The proof of the converse is omitted since it is easy to verify that a weighted Kendall distance satisfies Axioms II. □

The weighted Kendall distance provides a natural solution for the top-vs-

bottom issue. For instance, recall the example of ranking cities to live in, with

$$\begin{aligned}\pi &= (\text{Melbourne, Vienna, Vancouver, Toronto, Calgary,}\\ &\qquad \text{Adelaide, Sydney, Helsinki, Perth, Auckland}),\\ \pi' &= (\text{Melbourne, Vienna, Vancouver, Calgary, Toronto,}\\ &\qquad \text{Adelaide, Sydney, Helsinki, Perth, Auckland}),\\ \pi'' &= (\text{Vienna, Melbourne, Vancouver, Toronto, Calgary,}\\ &\qquad \text{Adelaide, Sydney, Helsinki, Perth, Auckland}),\end{aligned}$$

and choose the weight function $\varphi_{\langle i\ i+1\rangle} = 0.9^{i-1}$ for $i = 1, 2, \ldots, 9$. Then,

$$\mathsf{d}_\varphi(\pi, \pi') = 0.9^4 = 0.66 < \mathsf{d}_\varphi(\pi, \pi'') = 1,$$

as expected. In this case, we have chosen the weight function to be exponentially decreasing – the choice of the weight function in general depends on the application.

2.5 Computing the Weighted Kendall Distance

Computing the weighted Kendall distance between two permutations for an arbitrary weight function is not as straightforward a task as computing the Kendall τ distance. While an efficient algorithm for computing the weighted Kendall distance with respect to a general weight functions is not known, for an important class of weight functions – termed "monotonic" weight functions – the weighted Kendall distance may be computed efficiently, as described in §2.5.1. The case of monotonic weight functions is particularly important since these weight functions adequately address the top-vs-bottom issue by assigning higher weights to the top of rankings. An example of a monotonic weight function is the exponential weight described at the end of the previous subsection. In §2.5.2, we find the weighted Kendall distance for weight functions with only two (identical) non-zero weights. The distance for general weight functions can be computed via algorithms that essentially find minimum weight paths in graphs, as described in §2.5.3. A 2-approximation algorithm for general weight functions is given in §2.5.4.

2.5.1 Monotonic Weight Functions

A weight function $\varphi : \mathbb{A}_n \to \mathbb{R}_{\geq 0}$ is *decreasing* if $i > j$ implies that $\varphi_{\langle i\ i+1\rangle} \leq \varphi_{\langle j\ j+1\rangle}$. *Increasing* weight functions are defined similarly. A weight function is *monotonic* if it is increasing or decreasing.

Suppose $\tau \in \mathbb{A}(\pi, \sigma)$. The transform τ may be viewed as a sequence of moves that take each $i \in [n]$, from positions $\pi^{-1}(i)$ to position $\sigma^{-1}(i)$. Let the *walk*

followed by element i while moved by the transform τ be denoted by $p^{i,\tau} = \left(p_1^{i,\tau}, \ldots, p_{|p^{i,\tau}|+1}^{i,\tau}\right)$, where $\left|p^{i,\tau}\right|$ is the length of the walk $p^{i,\tau}$.

For example, consider

$$\begin{aligned}\pi &= (3,2,4,1),\\ \sigma &= (1,2,3,4),\\ \tau &= (\tau_1,\tau_2,\tau_3,\tau_4)\\ &= (\langle 3\ 4\rangle, \langle 2\ 3\rangle, \langle 1\ 2\rangle, \langle 2\ 3\rangle)\end{aligned}$$

and note that $\sigma = \pi\tau_1\tau_2\tau_3\tau_4$. We have

$$\begin{aligned}p^{1,\tau} &= (4,3,2,1),\\ p^{2,\tau} &= (2,3,2),\\ p^{3,\tau} &= (1,2,3),\\ p^{4,\tau} &= (3,4).\end{aligned}$$

We first bound the lengths of the walks $p^{i,\tau}, i \in [n]$. Let $I_i(\pi,\sigma)$ be the set consisting of elements $j \in [n]$ such that π and σ disagree on the pair $\{i,j\}$. In the transform τ, all elements of $I_i(\pi,\sigma)$ must be swapped with i by some $\tau_k, k \in [|\tau|]$. Each such swap contributes length one to the total length of the walk $p^{i,\tau}$ and thus, $\left|p^{i,\tau}\right| \geq |I_i(\pi,\sigma)|$.

As before, let d_φ denote the weighted Kendall distance with weight function φ. Since for any $\tau \in \mathbb{A}(\pi,\sigma)$,

$$\sum_{i=1}^{|\tau|} \varphi_{\tau_i} = \sum_{i=1}^{n} \frac{1}{2} \sum_{j=1}^{|p^{i,\tau}|} \varphi_{\left\langle p_j^{i,\tau}\ p_{j+1}^{i,\tau}\right\rangle},$$

we have

$$\mathsf{d}_\varphi(\pi,\sigma) = \min_{\tau\in\mathbb{A}(\pi,\sigma)} \sum_{i=1}^{n} \frac{1}{2} \sum_{j=1}^{|p^{i,\tau}|} \varphi_{\left\langle p_j^{i,\tau}\ p_{j+1}^{i,\tau}\right\rangle}.$$

Thus,

$$\mathsf{d}_\varphi(\pi,\sigma) \geq \sum_{i=1}^{n} \frac{1}{2} \min_{p^i\in P_i(\pi,\sigma)} \sum_{j=1}^{|p^i|} \varphi_{\left\langle p_j^i\ p_{j+1}^i\right\rangle}, \tag{2.10}$$

where for each i, $P_i(\pi,\sigma)$ denotes the set of walks of length $|I_i(\pi,\sigma)|$, starting from $\pi^{-1}(i)$ and ending in $\sigma^{-1}(i)$. Let

$$p^{i,\star}(\pi,\sigma) = \arg\min_{p^i\in P_i(\pi,\sigma)} \sum_{j=1}^{|p^i|} \varphi_{\left\langle p_j^i\ p_{j+1}^i\right\rangle}$$

be the minimum weight walk from $\pi^{-1}(i)$ to $\sigma^{-1}(i)$ with length $|I_i(\pi,\sigma)|$. If clear from the context, we write $p^{i,\star}(\pi,\sigma)$ as $p^{i,\star}$.

We show next that for decreasing weight functions, the bound given in (2.10) is achievable and thus the value on the right side gives the weighted Kendall

Algorithm 1 FINDTAUMONOTONE

Input: $\pi, \sigma \in \mathbb{S}_n$
Output: $\tau^\star = \arg\min_{\tau \in A(\pi,\sigma)} \mathrm{wt}(\tau)$
$\pi_0 \leftarrow \pi$
$t \leftarrow 0$
for $r = \sigma(1), \sigma(2), \ldots, \sigma(n)$ **do**
    while $\pi_t^{-1}(r) > \sigma^{-1}(r)$ **do**
        $\tau^\star_{t+1} \leftarrow \left(\pi_t^{-1}(r) - 1 \ \pi_t^{-1}(r)\right)$
        $\pi_{t+1} \leftarrow \pi_t \tau^\star_{t+1}$
        $t \leftarrow t + 1$

distance for this class of weight functions.

Consider $\pi, \sigma \in \mathbb{S}_n$ and a decreasing weight function φ. For each i, it follows that $p^{i,\star}(\pi, \sigma)$ extends to positions with largest possible indices, i.e., $p^{i,\star} = (\pi^{-1}(i), \cdots, \ell_i - 1, \ell_i, \ell_i - 1, \ldots, \sigma^{-1}(i))$ where ℓ_i is the solution to the equation

$$\ell_i - \pi^{-1}(i) + \ell_i - \sigma^{-1}(i) = |I_i(\pi, \sigma)|$$

and thus $\ell_i = \left(\pi^{-1}(i) + \sigma^{-1}(i) + |I_i(\pi, \sigma)|\right)/2$.

We show next that there exists a transform $\tau^\star$ with $p^{i,\tau^\star} = p^{i,\star}$, and so equality in (2.10) can be achieved. The transform is described in Algorithm 1.

The transform in question, $\tau^\star$, converts π into σ in n rounds. In Algorithm 1, the variable r takes values $\sigma(1), \sigma(2), \ldots, \sigma(n)$, in that given order. For each value of r, $\tau^\star$ moves r through a sequence of adjacent transpositions from its current position in π_t, $\pi_t^{-1}(r)$, to position $\sigma^{-1}(r)$.

Fix $i \in [n]$. For values of r such that $\sigma^{-1}(r) < \sigma^{-1}(i)$, i is swapped with r via an adjacent transposition if $\pi^{-1}(r) > \pi^{-1}(i)$. For $r = i$, i is swapped with all elements k such that $\pi^{-1}(k) < \pi^{-1}(i)$ and $\sigma^{-1}(i) < \sigma^{-1}(k)$. For r such that $\sigma^{-1}(r) > \sigma^{-1}(i)$, i is not swapped with other elements. Hence, i is swapped precisely with elements of the set $I_i(\pi, \sigma)$ and thus, $|p^{i,\tau^\star}(\pi, \sigma)| = |I_i(\pi, \sigma)|$. Furthermore, it can be seen that, for each i, $p^{i,\tau^\star}(\pi, \sigma) = (\pi^{-1}(i), \ldots, \ell_i' - 1, \ell_i', \ell_i' - 1, \ldots, \sigma^{-1}(i))$, for some ℓ_i'. Since $|p^{i,\tau^\star}(\pi, \sigma)| = |I_i(\pi, \sigma)|$, ℓ_i' also satisfies the equation

$$\ell_i' - \pi^{-1}(i) + \ell_i' - \sigma^{-1}(i) = |I_i(\pi, \sigma)|,$$

implying that $\ell_i' = \ell_i$ and thus $p^{i,\tau^\star} = p^{i,\star}$. Consequently, one has the following result.

Proposition 2.10. *For rankings $\pi, \sigma \in \mathbb{S}_n$, and a decreasing weighted Kendall weight function φ, we have*

$$\mathsf{d}_\varphi(\pi, \sigma) = \sum_{i=1}^{n} \frac{1}{2} \left(\sum_{j=\pi^{-1}(i)}^{\ell_i - 1} \varphi_{\langle j \ j+1 \rangle} + \sum_{j=\sigma^{-1}(i)}^{\ell_i - 1} \varphi_{\langle j \ j+1 \rangle} \right)$$

where $\ell_i = \left(\pi^{-1}(i) + \sigma^{-1}(i) + |I_i(\pi, \sigma)|\right)/2$.

Increasing weight functions may be analyzed similarly.

Example 2.11. Consider the rankings $\pi = (4,3,1,2)$ and $e = (1,2,3,4)$ and a decreasing weight function φ. We have $|I_i(\pi,e)| = 2$ for $i = 1,2$ and $|I_i(\pi,e)| = 3$ for $i = 3,4$. Furthermore,

$$\begin{aligned}
\ell_1 &= \frac{3+1+2}{2} = 3, & p^{1,\star} &= (3,2,1),\\
\ell_2 &= \frac{4+2+2}{2} = 4, & p^{2,\star} &= (4,3,2),\\
\ell_3 &= \frac{2+3+3}{2} = 4, & p^{3,\star} &= (2,3,4,3),\\
\ell_4 &= \frac{1+4+3}{2} = 4, & p^{4,\star} &= (1,2,3,4).
\end{aligned}$$

The minimum weight transform is

$$\tau^\star = \left(\underbrace{\langle 3\ 2\rangle, \langle 2\ 1\rangle}_{1}, \underbrace{\langle 4\ 3\rangle, \langle 3\ 2\rangle}_{2}, \underbrace{\langle 4\ 3\rangle}_{3} \right),$$

where the numbers under the braces denote the value r corresponding to the indicated transpositions. The distance between π and e is

$$\mathsf{d}_\varphi(\pi,e) = \varphi_{\langle 1\ 2\rangle} + 2\varphi_{\langle 2\ 3\rangle} + 2\varphi_{\langle 3\ 4\rangle}.$$

Example 2.12. The bound given in (2.10) is not tight for general weight functions as seen in this example. Consider $\pi = (4,2,3,1)$, $\sigma = (1,2,3,4)$, and a weight function φ with $\varphi_{\langle 1\ 2\rangle} = 2$, $\varphi_{\langle 2\ 3\rangle} = 1$, and $\varphi_{\langle 3\ 4\rangle} = 2$. Note that the domain of φ is the set of adjacent transpositions. We have

$$\begin{aligned}
p^{1,\star} &= (4,3,2,1),\\
p^{2,\star} &= (2,3,2),\\
p^{3,\star} &= (3,2,3),\\
p^{4,\star} &= (1,2,3,4).
\end{aligned}$$

Suppose that a transform $\tau \in \mathbb{A}(\pi,\sigma)$ exists such that $p^{i,\star} = p^{i,\tau}, i = 1,2,3,4$. From $p^{i,\star}$, it follows that in τ, transpositions $\langle 1\ 2\rangle$ and $\langle 3\ 4\rangle$ each appear once and $\langle 2\ 3\rangle$ appears twice. It can be shown, by considering all possible re-orderings of $\{\langle 1\ 2\rangle, \langle 1\ 2\rangle, \langle 2\ 3\rangle, \langle 2\ 3\rangle, \langle 2\ 3\rangle\}$, or by an application of Lemma 3.5 of the next chapter, that τ does not transform π into σ. Hence, for this example, the lower bound (2.10) is not achievable.

2.5.1.1 Average Distance for Decreasing Weights

The Kendall τ distance between two rankings may be viewed in the following way: each pair of candidates for which the two rankings disagree contribute one

unit to the distance between the rankings. Owing to Algorithm 1, the weighted Kendall distance with a decreasing weight function can be regarded in a similar manner: each pair of candidates for which the two rankings disagree contributes $\varphi_{\langle s\ s+1\rangle}$, for some s, to the distance between the rankings.

Consider a pair a and b such that $\pi^{-1}(b) < \pi^{-1}(a)$ and $\sigma^{-1}(a) < \sigma^{-1}(b)$. In Algorithm 1, there exists a transposition $\tau_t^\star = \langle s\ s+1\rangle$ that swaps a and b where

$$s = \pi^{-1}(b) + \left|\left\{k : \sigma^{-1}(k) < \sigma^{-1}(a), \pi^{-1}(k) > \pi^{-1}(b)\right\}\right|,$$

that is, s equals the sum of $\pi^{-1}(b)$ and the number of elements that appear before a in σ and after b in π. It is not hard to see that s can also be written in a way that is symmetric with respect to π and σ, as

$$\begin{aligned} s &= \pi^{-1}(b) + \sigma^{-1}(a) - \left|\left\{k : \pi^{-1}(k) < \pi^{-1}(b), \sigma^{-1}(k) < \sigma^{-1}(a)\right\}\right| - 1 \\ &= n - 1 - \left|\left\{k : \pi^{-1}(k) > \pi^{-1}(b), \sigma^{-1}(k) > \sigma^{-1}(a)\right\}\right|. \end{aligned}$$

As an example, consider $\varphi_{\langle i\ i+1\rangle} = n - i$. Then,

$$\begin{aligned} \mathsf{d}_\varphi(\pi,\sigma) &= \sum_{(b,a)\in\mathscr{I}(\pi,\sigma)} \left(1 + \left|\left\{k : \pi^{-1}(k) > \pi^{-1}(b), \sigma^{-1}(k) > \sigma^{-1}(a)\right\}\right|\right) \\ &= \mathsf{d}_K(\pi,\sigma) + \sum_{(b,a)\in\mathscr{I}(\pi,\sigma)} \left|\left\{k : \pi^{-1}(k) > \pi^{-1}(b), \sigma^{-1}(k) > \sigma^{-1}(a)\right\}\right| \end{aligned}$$

where $\mathscr{I}(\pi,\sigma)$ is the set of ordered pairs (b,a) such that $\pi^{-1}(b) < \pi^{-1}(a)$ and $\sigma^{-1}(a) < \sigma^{-1}(b)$. Note that the weighted Kendall distance d_φ equals the Kendall τ distance plus a sum that captures the influence of assigning higher importance to the top positions of the rankings.

These observations allow us to easily compute the expected value of the distance between the identity permutation and a randomly and uniformly chosen permutation $\pi \in \mathbb{S}_n$. For $1 \le a < b \le n$ and $s \in [n-1]$, let X_{ab}^s be an indicator variable that equals one if and only if $\pi^{-1}(a) > \pi^{-1}(b)$ and

$$\left|\left\{k > a : \pi^{-1}(k) > \pi^{-1}(b)\right\}\right| = n - 1 - s.$$

The expected distance between the two permutations equals

$$E[\mathsf{d}_\varphi(\pi, e)] = \sum_{s=1}^{n-1} \varphi_{\langle s\ s+1\rangle} \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} E\left[X_{ab}^s\right]. \tag{2.11}$$

By the definition of X_{ab}^s, $E\left[X_{ab}^s\right]$ equals the probability of the event that $n-1-s$ elements of $\{a+1,\ldots,n\}\backslash\{b\}$ and a appear after b in π. There are $\binom{n-a-1}{n-s-1}$ ways to choose $n-s-1$ elements from $\{a+1,\ldots,n\}\backslash\{b\}$, $\binom{n}{a-1}(a-1)!$ ways to assign positions to the elements of $\{1,2,\ldots,a-1\}$, $(s-a)!$ ways to arrange the $s-a$ elements of $\{a+1,\ldots,n\}\backslash\{b\}$ that appear before b, and $(n-s)!$ ways to arrange

a and the $n-1-s$ elements of $\{a+1,\ldots,n\}\backslash\{b\}$ that appear after b. Hence,

$$\begin{aligned} E[X_{ab}^s] &= \frac{1}{n!}\binom{n-a-1}{n-s-1}\binom{n}{a-1}(a-1)!(s-a)!(n-s)! \\ &= \frac{n-s}{(n-a+1)(n-a)}, \end{aligned}$$

for $1 \le a \le s$, and $E[X_{ab}^s] = 0$ for $a > s$. Using this expression in (2.11), we obtain

$$\begin{aligned} E[\mathsf{d}_\varphi(e,\pi)] &= \sum_{s=1}^{n-1} \varphi_{\langle s\ s+1\rangle} \sum_{a=1}^{s} \frac{n-s}{n-a+1} \\ &= \sum_{s=1}^{n-1} \varphi_{\langle s\ s+1\rangle}(n-s)(H_n - H_{n-s}), \end{aligned}$$

where $H_i = \sum_{l=1}^{i} \frac{1}{l}$. Indeed, for $\varphi_{\langle s\ s+1\rangle} = 1, s \in [n-1]$, we recover the well known result that

$$\begin{aligned} E[\mathsf{d}_K(e,\pi)] &= \sum_{s=1}^{n-1} (n-s)(H_n - H_{n-s}) \\ &= \sum_{k=1}^{n-1} k(H_n - H_k) \\ &= \frac{1}{2}\binom{n}{2}. \end{aligned}$$

For $\varphi_{\langle s\ s+1\rangle} = n-s$, the average distance equals

$$\begin{aligned} E[\mathsf{d}_\varphi(e,\pi)] &= \sum_{s=1}^{n-1} (n-s)^2(H_n - H_{n-s}) \\ &= \sum_{k=1}^{n-1} k^2(H_n - H_k) \\ &= \frac{1}{2}\binom{n}{2} + \frac{2}{3}\binom{n}{3}. \end{aligned}$$

2.5.2 Weight Functions with Two Identical Non-zero Weights

Another example of a weighted Kendall τ distance for which a closed form solution may be found is described below.

For a pair of integers a and b, where $1 \le a < b < n$, define the weight function φ as:

$$\varphi_{\langle i\ i+1\rangle} = \begin{cases} 1, & i \in \{a,b\} \\ 0, & \text{else.} \end{cases} \tag{2.12}$$

Such weight functions may be used in voting problems where one only penalizes moving a link from one page (say, top-ten page) to another page (say, eleven-to-twenty page). In other words, one only penalizes moving an item from a "high-ranked" set of positions to "average-rank" or "low-rank" positions.

Let $R_1 = \{1, \ldots, a\}$, $R_2 = \{a+1, \ldots, b\}$, and $R_3 = \{b+1, \ldots, n\}$, and define

$$N_{ij}^{\pi} = |\{k \in R_j : \pi^{-1}(k) \in R_i\}|, \quad i, j \in \{1, 2, 3\}.$$

In §3.4.3, we show that

$$\mathsf{d}_{\varphi}(\pi, e) = \begin{cases} 2N_{13}^{\pi} + N_{12}^{\pi} + N_{23}^{\pi}, & \text{if } N_{21}^{\pi} \geq 1 \text{ or } N_{23}^{\pi} \geq 1, \\ 2N_{13}^{\pi} + 1, & \text{if } N_{21}^{\pi} = N_{23}^{\pi} = 0. \end{cases}$$

In the next section, we present algorithms for computing and approximating the weighted Kendall distance with general weight functions.

2.5.3 General Weight Functions

The results of §2.5.1 imply that at least for one class of weight functions that capture the importance of the top entries in a ranking, computing the weighted Kendall distance has time complexity $O(n^2)$. Hence, distance computation efficiency does not represent a bottleneck for the employment of this form of the weighted Kendall distance.

Next, we discuss two algorithms for computing the *exact* weighted Kendall distance. While the exact computation has super exponential time complexity, for a small number of candidates – say, less than 10 – the computation can be performed in reasonable time. A small number of candidates and a large number of voters are frequently encountered in social choice applications, but less frequently in computer science.

As already pointed out, the Kendall τ and the weighted Kendall distance are graphic distances. In the latter case, we define a graph G with vertex set indexed by $\mathbb{S}_n$ and an edge of weight $\varphi_{\langle i\ i+1\rangle}$ between each pair of vertices π and σ whenever there is an i such that $\pi = \sigma\langle i\ i+1\rangle$. The numbers of vertices and edges of G are $|V| = n!$ and $|E| = n!(n-1)/2$, respectively. Dijkstra's algorithm with Fibonacci heaps [55] for finding the minimum weight path in a graph provides the distances of all $\pi \in \mathbb{S}_n$ to the identity in time $O(|E| + |V| \log |V|) = O(n!\, n \log n)$.

The complexity of the algorithm for finding the distance between $\pi \in \mathbb{S}_n$ and the identity may be actually shown to be $O(n(\mathsf{d}_K(\pi, e))!)$, which is significantly smaller than $\Omega(n!)$ for permutations at small Kendall τ distance. To see this, consider the following observation. For π in $\mathbb{S}_n$, there exists a transform $\tau = (\tau_1, \ldots, \tau_m)$ of minimum weight that transforms π into e, such that $m = \mathsf{d}_K(\pi, e)$. In other words, each transposition of τ eliminates one inversion when transforming π into e. Hence, $\pi\tau_1$ has one less inversion than π. As a result,

$$\mathsf{d}_{\varphi}(\pi, e) = \min_{i:\pi(i)>\pi(i+1)} \left(\varphi_{\langle i\ i+1\rangle} + \mathsf{d}_{\varphi}(\pi\langle i\ i+1\rangle, e)\right). \tag{2.13}$$

Suppose that computing the weighted Kendall distance between the identity and a permutation π, with $\mathsf{d}_K(\pi, e) = d$, can be performed in time T_d. From (2.13), we have

$$T_d = a\,n + d\,T_{d-1}, \quad \text{for } d \geq 2,$$

and $T_1 = a\,n$, for some constant a. By letting $U_d = T_d/(a\,n\,d!)$, we obtain $U_d = U_{d-1} + \frac{1}{d!}, d \geq 2$, and $U_1 = 1$. Hence, $U_d = \sum_{i=1}^{d} \frac{1}{i!}$. It can then be shown that $d!\,U_d = \lfloor d!(e-1) \rfloor$, and thus $T_d = a\,n\lfloor d!\,(e-1) \rfloor = O(nd!)$.

The expression (2.13) can also be used to find the distances of all $\pi \in \mathbb{S}_n$ from the identity by first finding the distances of permutations $\pi \in \mathbb{S}_n$ with $\mathsf{d}_K(\pi, e) = 1$, then finding the distances of permutations $\pi \in \mathbb{S}_n$ with $\mathsf{d}_K(\pi, e) = 2$, and so on.[3] Unfortunately, the average Kendall τ distance between a randomly chosen permutation and the identity is $\binom{n}{2}/2$ (see the derivation of this known and a related novel result regarding the weighted Kendall distance in §2.5.1.1), which limits the applicability of this algorithm to uniformly and randomly chosen votes.

2.5.4 Approximation for General Weight Functions

In what follows, we present a polynomial-time 2-approximation algorithm for computing the most general form of weighted Kendall distances.

In order to approximate the weighted Kendall distance, $\mathsf{d}_\varphi(\pi, \sigma)$, we use the function $D_\varphi(\pi, \sigma)$, defined as

$$D_\varphi(\pi, \sigma) = \sum_{i=1}^{n} w(\pi^{-1}(i) : \sigma^{-1}(i)),$$

where

$$w(k : l) = \begin{cases} \sum_{h=k}^{l-1} \varphi_{\langle h\ h+1 \rangle}, & \text{if } k < l, \\ \sum_{h=l}^{k-1} \varphi_{\langle h\ h+1 \rangle}, & \text{if } k > l, \\ 0, & \text{if } k = l, \end{cases}$$

denotes the sum of the weights of adjacent transpositions

$$\langle k\ k+1 \rangle, \langle k+1\ k+2 \rangle, \ldots, \langle l-1\ l \rangle$$

if $k < l$; the sum of the weights of adjacent transpositions

$$\langle l\ l+1 \rangle, \langle l+1\ l+2 \rangle, \ldots, \langle k-1\ k \rangle$$

if $l < k$; and 0 if $k = l$.

The following lemma states lower and upper bounds for d_φ in terms of D_φ. The lemma is useful in practice, since D_φ can be computed in time $O(n^2)$, and

[3] Note that such an algorithm requires that the set of permutations at a given Kendall τ distance from the identity be known.

provides the desired 2-approximation.

Lemma 2.13. *For a weighted Kendall weight function φ and for permutations π and σ,*

$$\frac{1}{2}D_\varphi(\pi,\sigma) \le \mathsf{d}_\varphi(\pi,\sigma) \le D_\varphi(\pi,\sigma).$$

We omit the proof of the lemma, since it follows from a more general result stated in the next chapter, and only remark that the lower-bound presented above lemma is weaker than the lower-bound given by (2.10).

2.6 Aggregation with Weighted Kendall Distances

In order to explain the ability of the weighted Kendall distance in addressing the top-vs-bottom aggregation issue, in what follows, we present a number of examples that illustrate how the choice of the weight function influences the final form of the aggregate. We focus on *decreasing weight functions* and compare our results to those obtained using the classical Kendall τ distance.

We refer to the problem given in (2.2) as the *aggregation problem*, and to a solution to the aggregation problem using the Kendall τ as a *Kemeny aggregate*. All the aggregation results are obtained via exhaustive search since the examples are small and only used for illustrative purposes. Aggregation is, in general, a hard problem and we postpone the analysis of the complexity of computing aggregate rankings, and aggregate approximation algorithms, until Chapter 4.

Example 2.14. Consider the set of rankings listed in Σ, where each row represents a ranking (vote),

$$\Sigma = \left(\begin{array}{ccccc} 4 & 1 & 2 & 5 & 3 \\ \hline 4 & 2 & 1 & 3 & 5 \\ \hline 1 & 4 & 5 & 2 & 3 \\ \hline 2 & 3 & 1 & 5 & 4 \\ \hline 5 & 3 & 1 & 2 & 4 \end{array}\right).$$

The Kemeny aggregate for this set of rankings is $(1,4,2,5,3)$. Note that despite the fact that candidate 4 was ranked twice at the top of the list – more than any other candidate – it is ranked only second in the aggregate. This may be attributed to the fact that 4 was ranked last by two voters.

Consider next the weight function φ with $\varphi_{\langle i\ i+1\rangle} = (2/3)^{i-1}, i \in [4]$. The optimum aggregate ranking for this weight equals $(4,1,2,5,3)$ which puts 4 before 1, similar to what the plurality rule would do.[4] The reason behind this swap is that φ emphasizes strong showings of a candidate and downplays its weak showings, since weak showings have a smaller effect on the distance as the weight function is decreasing. In other words, higher ranks are more important than lower ranks when determining the position of a candidate.

[4]In *plurality votes*, the candidate with the most first-place rankings is declared the winner.

Example 2.15. Consider the set of rankings listed in Σ,

$$\Sigma = \begin{pmatrix} 1 & 4 & 2 & 3 \\ \hline 1 & 4 & 3 & 2 \\ \hline 2 & 3 & 1 & 4 \\ \hline 4 & 2 & 3 & 1 \\ \hline 3 & 2 & 4 & 1 \end{pmatrix}.$$

The Kemeny aggregate is $(4, 2, 3, 1)$. Note that although the majority of voters prefer 1 to 4, 1 is ranked last and 4 is ranked first. More precisely, we observe that according to the pairwise majority test, 1 beats 4 but loses to 2 and 3. On the other hand, 4 is preferred to both 2 and 3 but, as mentioned before, loses to 1. Problems like this do not arise due to a weakness of Kemeny's approach, but due to the inherent "rational intractability" of rank aggregation. As stated by Arrow [56], for *any* "reasonable" rank aggregation method, there exists a set of votes such that the aggregated ranking prefers one candidate to another while the majority of voters prefer the later to the former.

Let us now focus on a weighted Kendall distance with weight function $\varphi_{\langle i\ i+1\rangle} = (2/3)^{i-1}, i = 1, 2, 3$. The optimal aggregate ranking for this distance equals $(1, 4, 2, 3)$. Again, we see a candidate with both strong showings and weak showings, candidate 1, beat a candidate with a rather average performance. Note that in this solution as well, there exist candidates for which the opinion of the majority is ignored: 1 is placed before 2 and 3, while according to the pairwise majority opinion it loses to both.

Example 2.16. Consider the set of rankings listed in Σ,

$$\Sigma = \begin{pmatrix} 5 & 4 & 1 & 3 & 2 \\ \hline 1 & 5 & 4 & 2 & 3 \\ \hline 4 & 3 & 5 & 1 & 2 \\ \hline 1 & 3 & 4 & 5 & 2 \\ \hline 4 & 2 & 5 & 3 & 1 \\ \hline 1 & 2 & 5 & 3 & 4 \\ \hline 2 & 4 & 3 & 5 & 1 \end{pmatrix}.$$

With the weight function $\varphi_{\langle i\ i+1\rangle} = (2/3)^{i-1}$ for $i \in [4]$, the aggregate equals $(4, 1, 5, 2, 3)$. The winner is 4, while the plurality rule winner is 1 as it appears three times on the top. Next, we increase the rate of decay of the weight function and let $\varphi_{\langle i\ i+1\rangle} = (1/3)^{i-1}, i \in [4]$. The solution now is $(1, 4, 2, 5, 3)$, and the winner is candidate 1, the same as the plurality rule winner. Note that the plurality winner is the aggregate based on the weighted Kendall distance with weight function $\varphi^{(p)}$,

$$\varphi^{(p)}_{\langle i\ i+1\rangle} = \begin{cases} 1, & i = 1, \\ 0, & \text{else.} \end{cases}$$

The Kemeny aggregate is $(4, 5, 1, 2, 3)$.

A shortcoming of distance-based rank aggregation is that sometimes the solution is not unique, and that the possible solutions differ widely. The following example describes one such scenario.

Example 2.17. Suppose that the votes are given by Σ,

$$\Sigma = \left(\begin{array}{ccc} 1 & 2 & 3 \\ \hline 1 & 2 & 3 \\ \hline 3 & 2 & 1 \\ \hline 2 & 1 & 3 \end{array} \right).$$

Here, the permutations $(1, 2, 3)$ and $(2, 1, 3)$ are the Kemeny aggregates, with cumulative distance 4 from Σ. When the Kemeny aggregate is not unique, it may be possible to obtain a unique solution by using a non-uniform weight function. In this example, it can be shown that for any non-uniform weight function φ with $\varphi_{\langle 1\ 2\rangle} > \varphi_{\langle 2\ 3\rangle}$, the solution is unique, namely, $(1, 2, 3)$.

A similar situation occurs if the last vote is changed to $(2, 3, 1)$. In that case, the permutations $(1, 2, 3)$, $(2, 1, 3)$, and $(2, 3, 1)$ are the Kemeny aggregates with cumulative distance 5 from Σ. Again, for any non-uniform weight function φ with $\varphi_{\langle 1\ 2\rangle} > \varphi_{\langle 2\ 3\rangle}$ the solution is unique and equal to $(1, 2, 3)$.

To summarize, the above examples illustrate how a proper choice for the weighted Kendall distance insures that top ranks are emphasized and how one may over-rule a moderate number of low rankings using a specialized distance formula. One may argue that certain generalizations of Borda's method, involving non-uniform gaps between ranking scores, may achieve similar goals. This is not the case, as will be illustrated in what follows.

One major difference between generalized Borda and weighted Kendall distances is in the *majority criterion* [57], which states that the candidate ranked first by the majority of voters has to be ranked first in the aggregate.[5] Borda's aggregate with an arbitrary score assignments does not have this property, while aggregates obtained via weighted Kendall distances with decreasing weights (not identically equal to zero) have this property.

We first show that the Borda method with a fixed, but otherwise arbitrary, set of scores may not satisfy the majority criterion. We prove this claim for $n = 3$. A similar argument can be used to establish this claim for $n > 3$.

Suppose, for simplicity, that the number m of voters is odd and that, for each vote, a score s_i is assigned to a candidate with rank i, $i = 1, 2, 3$. Here, we assume that $s_1 > s_2 > s_3 \geq 0$. Suppose also that $(m + 1)/2$ of the votes equal (a, b, c) and that $(m - 1)/2$ of the votes equal (b, c, a). Let the total Borda scores for

[5] Note that a candidate ranked first by the majority is a Condorcet candidate. It is desirable that an aggregation rule satisfy the majority criterion and indeed most do, including the Condorcet method, the plurality rule, the single transferable vote method, and the Coombs method.

candidates a and b be denoted by S and S', respectively. We have

$$S = \frac{m+1}{2}s_1 + \frac{m-1}{2}s_3,$$
$$S' = \frac{m+1}{2}s_2 + \frac{m-1}{2}s_1,$$

and thus $S - S' = s_1 - m\left(\frac{s_2-s_3}{2}\right) - \frac{s_2+s_3}{2}$. If $m > \frac{2s_1-(s_2+s_3)}{s_2-s_3}$, then $S - S' < 0$ and Borda's method ranks b higher than a. As a result, candidate a, ranked highest by more than half of the voters, is not ranked first according to Borda's rule. This is not the case with weighted Kendall distances, as shown below.

Proposition 2.18. *An aggregate ranking obtained using the weighted Kendall distance with a decreasing weight function not identically equal to zero satisfies the majority criterion.*

Proof. Suppose that the weight function is φ, and let $w_i = \varphi_{\langle i\ i+1\rangle}$. Since w is decreasing and not identically equal to zero, we have $w_1 > 0$. Let a_1 be a candidate that is ranked first by a majority of voters. Partition the set of votes into two sets, C and D, where C is the set of votes that rank a_1 first and D is the set of votes that do not. Furthermore, denote the aggregate ranking by π.

Suppose that a_1 is not ranked first in π and that π equals

$$(a_2, \ldots, a_i, a_1, a_{i+1}, \ldots, a_n),$$

for some $i \geq 2$. Let $\pi' = (a_1, a_2, \ldots, a_n)$. We show that

$$\sum_{j=1}^{m} \mathsf{d}_\varphi(\pi, \sigma_j) > \sum_{j=1}^{m} \mathsf{d}_\varphi(\pi', \sigma_j)$$

which contradicts the optimality of π. Hence, a_1 must be ranked first in π.

For $\sigma \in C$, we have

$$\mathsf{d}_\varphi(\pi, \sigma) = \mathsf{d}_\varphi(\pi, \pi') + \mathsf{d}_\varphi(\pi', \sigma). \tag{2.14}$$

To see the validity of this claim, note that if π is to be transformed to σ via Algorithm 1, it is first transformed to π' by moving a_1 to the first position.

For $\sigma \in D$, we have

$$\mathsf{d}_\varphi(\pi', \sigma) \leq \mathsf{d}_\varphi(\pi', \pi) + \mathsf{d}_\varphi(\pi, \sigma), \tag{2.15}$$

which follows from the triangle inequality.

To complete the proof, we write

$$\begin{aligned}
\sum_{j=1}^{m} \mathsf{d}_\varphi(\pi, \sigma_j) &= \sum_{\sigma \in C} \mathsf{d}_\varphi(\pi, \sigma) + \sum_{\sigma \in D} \mathsf{d}_\varphi(\pi, \sigma) \\
&\geq \sum_{\sigma \in C} \mathsf{d}_\varphi(\pi', \sigma) + |C| \mathsf{d}_\varphi(\pi, \pi') \\
&\quad + \sum_{\sigma \in C} \mathsf{d}_\varphi(\pi', \sigma) - |D| \mathsf{d}_\varphi(\pi, \pi') \\
&= \sum_{j=1}^{m} \mathsf{d}_\varphi(\pi', \sigma) + (|C| - |D|)\, \mathsf{d}_\varphi(\pi, \pi') \\
&> \sum_{j=1}^{m} \mathsf{d}_\varphi(\pi', \sigma)
\end{aligned}$$

where the first inequality follows from (2.14) and (2.15), and the second inequality follows from the facts that $|C| > |D|$ and that $\mathsf{d}_\varphi(\pi, \pi') \geq w_1 > 0$.

□

Chapter 3

Weighted Transposition Distance

3.1 Introduction

In some vote aggregation problems, the identity of the candidates may not be known. On the other hand, many applications require that the identity of the candidates be revealed. In this case, candidates are frequently evaluated in terms of some similarity criteria – for example, area of expertise, gender, working hour schedule etc. Hence, pairs of candidates may have different degrees of similarity and swapping candidates that are similar should be penalized less than swapping candidates that are not similar according to the given ranking criteria. For example, in a faculty search ranking one may want to have at least one but not more than two physicists ranked among the top 10 candidates, or at least two women among the top 5 candidates.

As in Chapter 2, consider the Economist Intelligence Unit ranking of the "World's 10 best cities to live in":

$$\pi = (\text{Melbourne, Vienna, Vancouver, Toronto, Calgary,}$$
$$\text{Adelaide, Sydney, Helsinki, Perth, Auckland}).$$

The following two rankings are obtained from π by one swap of adjacent entries:

$$\pi' = (\text{Melbourne, Vienna, Vancouver, Calgary, Toronto,}$$
$$\text{Adelaide, Sydney, Helsinki, Perth, Auckland}),$$
$$\pi'' = (\text{Vienna, Melbourne, Vancouver, Toronto, Calgary,}$$
$$\text{Adelaide, Sydney, Helsinki, Perth, Auckland}).$$

It may be observed that the swap in π'' involves cities on two different continents, which may shift the general opinion about the cities' countries of origin. On the other hand, the two cities swapped in π' are both in Canada, so that the swap is not likely to change the perception of quality of living in that country. This points to the need for distance measures that take into account similarities and dissimilarities among candidates.

This chapter contains material from [58], coauthored with C.-Y. Chen, O. Milenkovic, and N. Kashyap; and from [59, 60], coauthored with O. Milenkovic.

Similarity of items may be captured by assigning weights to swaps, and choosing the transposition weights so that swapping dissimilar items induces a higher distance compared to swapping similar items. We illustrate the concept of distance measures taking into account similarities via rankings of cities. For simplicity, we use shorter rankings than the ones mentioned above. Suppose that four cities: Melbourne, Sydney, Helsinki, and Vienna are ranked based on a certain set criteria as

$$\pi = (\text{Helsinki, Sydney, Vienna, Melbourne}), \tag{3.1}$$

and according to another set of criteria as

$$\sigma = (\text{Melbourne, Vienna, Helsinki, Sydney}). \tag{3.2}$$

The distance between π and σ is defined as follows. We assign weights to swapping the cities in the rankings, e.g., suppose that the weight of swapping cities in the same country is 1, on the same continent 2, and 3 otherwise. The *similarity distance* between π and σ is the minimum weight of any sequence of swaps of cities that convert π into σ. By inspection, or by methods discussed in this chapter, one can see that the similarity distance between π and σ equals 6. One of the sequences of swaps of weight 6 is as follows: first swap Helsinki and Sydney with weight 3, then swap Melbourne and Sydney with weight 1, and finally swap Vienna and Helsinki with weight 2.

Another application of the similarity distance is in the context of *assignment aggregation*, which is closely related to rank aggregation. Consider a committee with m members that is tasked with filling n jobs with n candidates. Suppose that each committee member provides a full assignment of candidates to jobs. The task is to aggregate the assignments given by individual committee members into one assignment. If a measure of similarity between the candidates is available, one can use similarity distance to aggregate the assignment by finding the median of the assignments provided by the committee members.

Note that in the definition of the similarity distance, the weights are based on the identity of the candidates being swapped, while in Chapter 2, the definition of the weighted Kendall distance relied on weights *based on the ranks of candidates being swapped*. It is straightforward to see that all statements made about swapping elements i and j may be converted to statements made about swapping elements at positions i and j by using the inverse of the ranking (permutation). To be consistent with Chapter 2, we define the *weighted transposition distance*, which is equivalent to the similarity distance when applied to inverse of rankings, and present the results of this chapter in terms of the weighted transposition distance. The relationship between the similarity distance and the weighted transposition distance is explained in detail in the next section.

The rest of this chapter is organized as follows. In §3.2, we present formal definitions and preliminaries useful for our analysis. Algorithms for computing the weighted transposition distance between a transposition and the identity are given in §3.3. In §3.4, we study the weighted transposition distance between permutations. We present approximation methods for general weight functions as well as exact algorithms for some special weight functions. In §3.5, we present an algorithm for finding the minimum weight transform among those with shortest length. We also show that the minimum weight, minimum length transform has weight that is not more than twice the weighted transposition distance. Finally, in §3.6, we demonstrate the application of the weighted transposition distance to rank aggregation using some examples.

3.2 Definitions and Preliminaries

To define the similarity distance formally, we write the rankings as permutations. To do so, we represent the n candidates using numbers 1 through n. For example, rankings given in (3.1) and (3.2) can be written as permutations by representing Melbourne by 1, Sydney by 2, Vienna by 3, and Helsinki by 4. This is equivalent to assuming that the identity ranking is

$$e = (\text{Melbourne, Sydney, Vienna, Helsinki}).$$

We then have $\pi = (4, 2, 3, 1)$ and $\sigma = (1, 4, 2, 3)$. The similarity information is represented via a *weight function* $\varphi : \mathbb{T}_n \to \mathbb{R}_{\geq 0}$, which assigns a nonnegative weight $\varphi_{\langle i\ j\rangle}$ to each transposition $\langle i\ j\rangle$. The weight function φ, representing the geographical similarity of the cities as discussed above, equals

$$\begin{aligned} \varphi_{\langle 1\ 2\rangle} &= 1, \quad \varphi_{\langle 1\ 3\rangle} = 3, \quad \varphi_{\langle 1\ 4\rangle} = 3 \\ \varphi_{\langle 2\ 3\rangle} &= 3, \quad \varphi_{\langle 2\ 4\rangle} = 3, \quad \varphi_{\langle 3\ 4\rangle} = 2. \end{aligned}$$

For permutations μ and ω, let $\mathbb{T}'(\mu, \omega)$ be the set of sequences $\tau = (\tau_1, \tau_2, \ldots, \tau_{|\tau|})$ of transpositions such that $\tau_{|\tau|} \cdots \tau_1 \mu = \omega$. Note that multiplication by a transposition on the left is equivalent to swapping the corresponding elements. For example, $\langle 2\ 3\rangle \pi = (4, 3, 1, 2)$. The weight of a sequence of transpositions is defined as the sum of the weights of its transpositions. That is, the weight of the sequence $\tau = (\tau_1, \ldots, \tau_{|\tau|})$ of transpositions equals

$$\mathrm{wt}(\tau) = \sum_{i=1}^{|\tau|} \varphi_{\tau_i},$$

where the empty sum is assigned value 0. The similarity distance between π and σ, denoted by $\mathsf{d}'_\varphi(\pi,\sigma)$, reads as

$$\mathsf{d}'_\varphi(\pi,\sigma) = \min_{\tau\in\mathbb{T}'(\pi,\sigma)} \mathrm{wt}(\tau).$$

Next, we define the weighted transposition distance between permutations π and σ, which is equal to the similarity distance applied to π^{-1} and σ^{-1}.

Definition 3.1. The *weighted transposition distance* between two permutations π and σ, with respect to a weight function φ, is defined as the minimum weight of a sequence $\tau = (\tau_1,\ldots,\tau_{|\tau|})$ of transpositions such that $\sigma = \pi\tau_1\cdots\tau_{|\tau|}$. This distance is denoted by d_φ. We refer to such a sequence of transpositions as a *(transposition) transform converting π into σ* and use $\mathbb{T}(\pi,\sigma)$ to denote the set of transforms that convert π into σ.

With this notation at hand, the weighted transposition distance between π and σ may be written as

$$\mathsf{d}_\varphi(\pi,\sigma) = \min_{\tau\in\mathbb{T}(\pi,\sigma)} \mathrm{wt}(\tau).$$

Since $\tau_{|\tau|}\cdots\tau_2\tau_1\pi = \sigma$ if and only if $\pi^{-1}\tau_1\tau_2\cdots\tau_{|\tau|} = \sigma^{-1}$, we have $\mathbb{T}'(\pi,\sigma) = \mathbb{T}(\pi^{-1},\sigma^{-1})$ and thus

$$\mathsf{d}'_\varphi(\pi,\sigma) = \mathsf{d}_\varphi(\pi^{-1},\sigma^{-1}).$$

As our discussion is not dependent on the particular choices for π and σ, in order to find the similarity distance d'_φ it suffices to find the weighted transposition distance d_φ. Throughout the rest of this chapter, we mainly focus on the weighted transposition distance.

For every nonnegative weight function, the weighted transposition distance d_φ is a pseudo-metric. That is, for every $\pi,\sigma,\omega\in\mathbb{S}_n$,

- $\mathsf{d}_\varphi(\pi,\pi) = 0$,
- $\mathsf{d}_\varphi(\pi,\sigma) \geq 0$,
- $\mathsf{d}_\varphi(\pi,\sigma) = \mathsf{d}_\varphi(\sigma,\pi)$,
- $\mathsf{d}_\varphi(\pi,\sigma) \leq \mathsf{d}_\varphi(\sigma,\omega) + \mathsf{d}_\varphi(\omega,\pi)$.

Furthermore, d_φ is left-invariant since $\mathbb{T}(\pi,\sigma) = \mathbb{T}(\omega\pi,\omega,\sigma)$.

The weighted transposition distance may be viewed as a generalization of the Kendall *tau* distance and the weighted Kendall distance: The definition of the Kendall τdistance and the weighted Kendall distance is based on transforming one permutation into another using *adjacent* transpositions. If instead all transpositions are allowed – including non-adjacent transpositions – the resulting distance is the weighted transposition distance. To obtain the Kendall τ

distance, let

$$\varphi_\theta = \begin{cases} 1, & \theta = \langle i\ i+1 \rangle, i \in [n-1] \\ \infty, & \text{else,} \end{cases}$$

and to obtain the weighted Kendall distance, let

$$\varphi_\theta = \begin{cases} w_i, & \theta = \langle i\ i+1 \rangle, i \in [n-1] \\ \infty, & \text{else,} \end{cases}$$

for a nonnegative function w.

For a given weight function φ, we let K_φ denote a complete undirected weighted graph with vertex set $[n]$, where the weight of each edge (ij) equals the weight $\varphi_{\langle i\ j \rangle}$ of the transposition $\langle i\ j \rangle$.

For a multigraph H with the same vertex set as that of K_φ, the weight of each edge (ij) of H is $\varphi_{\langle i\ j \rangle}$ and the weight of H is

$$\mathrm{wt}(H) = \sum_{(ij) \in E(H)} \varphi_{\langle i\ j \rangle},$$

that is, the sum of the weights of its edges.

For $\pi, \sigma \in \mathbb{S}_n$, let $p^*_\varphi(a, b)$ denote the minimum weight path from a to b in K_φ and define $D_\varphi(\pi, \sigma)$ as[1]

$$D_\varphi(\pi, \sigma) = \sum_{i=1}^{n} \mathrm{wt}\left(p^*_\varphi(\pi^{-1}(i), \sigma^{-1}(i))\right).$$

It is easy to verify that D_φ is a pseudo-metric. Furthermore, it is left-invariant since

$$\begin{aligned} D_\varphi(\pi, \sigma) &= \sum_{i=1}^{n} \mathrm{wt}\left(p^*_\varphi(\pi^{-1}(i), \sigma^{-1}(i))\right) \\ &= \sum_{i=1}^{n} \mathrm{wt}\left(p^*_\varphi(\pi^{-1}(\omega^{-1}(i)), \sigma^{-1}(\omega^{-1}(i)))\right) \\ &= D_\varphi(\omega\pi, \omega\sigma), \end{aligned}$$

for every $\pi, \sigma, \omega \in \mathbb{S}_n$.

3.2.1 Transpositions

The *predecessor* of $a \in [n]$ in a permutation π is $\pi^{-1}(a)$ and the *successor* of a is $\pi(a)$. There is an edge from the predecessor of a to a and an edge from a to its successor in the functional digraph of π, $\mathrm{dig}(\pi)$, defined in Chapter 1.

In addition to functional digraphs, a permutation can be represented using a binning scheme (bins and balls model). Consider a permutation $\pi \in \mathbb{S}_n$, n balls

[1] Note that this definition is consistent with the definition of a specialization of this function, given in Proposition 16.

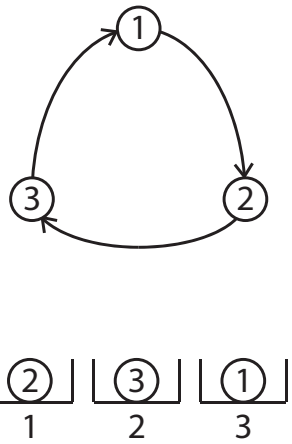


Figure 3.1: The functional digraph and the balls and bins model for permutation $\pi = (2, 3, 1)$.

labeled from 1 to n, and n bins, labeled in the same way. Ball j is placed in bin i if $\pi(i) = j$. An example is shown in Figure 3.1.

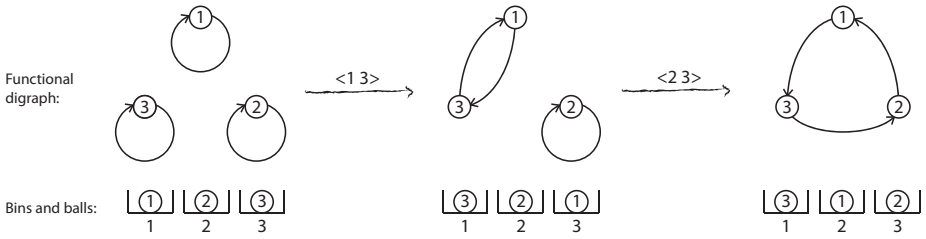
Multiplication of a permutation by a transposition on the left and on the right results in different permutations. For a permutation π, the product $\pi \langle a\ b \rangle$ is obtained by swapping elements in positions a and b in π. In the functional digraph, this is reflected by swapping the successors of a and b. In the binning model, the balls in the bins a and b are swapped. The product $\langle a\ b \rangle \pi$ is, however, obtained by swapping elements a and b in π. In the functional digraph, multiplication on the left is reflected by swapping the predecessors of a and b and in the binning model, by swapping the balls a and b. Figure 3.2 provides an example of these concepts. By *applying a transposition to a permutation* we mean that the permutation is multiplied by the transposition on the right unless stated otherwise.

For a sequence of transpositions $\tau = (\tau_1, \ldots, \tau_{|\tau|})$, with $\tau_i = \langle a_i\ b_i \rangle$, the multigraph of τ, denoted by $\mathrm{mgr}(\tau)$, is a multigraph with vertex set $[n]$ and edges $(a_i b_i)$ for each transposition τ_i. An example is shown in Figure 3.3. Note that two distinct sequences of transpositions may have the same multigraph.

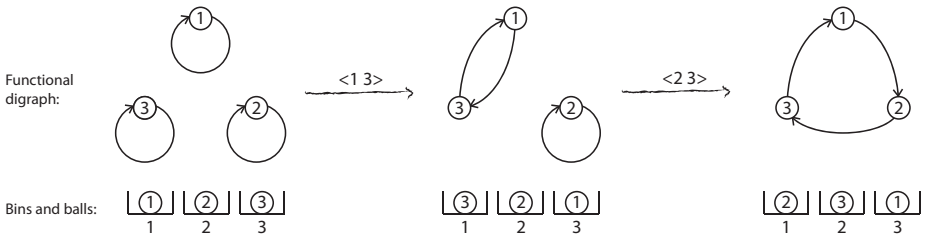
As a simple application of the balls and bins model, we state the following straightforward lemma.

Lemma 3.2. *For a permutation π and a transform $\tau \in \mathbb{T}(\pi, e)$, there exists a path from i to $\pi(i)$ in $\mathrm{mgr}(\tau)$ for each $i \in [n]$.*

Proof. Since $\tau \in \mathbb{T}(\pi, e)$, we have $e = \pi\tau_1 \cdots \tau_{|\tau|}$. Suppose $\tau_j = \langle x_j\ y_j \rangle$ and $|\tau| = k$. Consider the balls and bins representation where each bin is located on the vertex of $\mathrm{mgr}(\tau)$ with the same label, and initially bin i is occupied by ball $\pi(i)$. Through the sequence $\tau_1, \ldots, \tau_k$ of transpositions, applied one by one, π is transformed into the identity permutation. In the identity permutation bin i is occupied by ball i.



(a) Multiplication by transpositions on the right.



(b) Multiplication by transpositions on the left.

Figure 3.2: Multiplication of a permutation by a transposition on the left and on the right result in possibly different permutations. In (a), at each step the permutation is multiplied by a transposition on the right:
$(1,2,3)\langle 1\ 3\rangle = (3,2,1)$ and $(3,2,1)\langle 2\ 3\rangle = (3,1,2)$. In (b), at each step the permutation is multiplied by a transposition on the left:
$\langle 1\ 3\rangle(1,2,3) = (3,2,1)$ and $\langle 2\ 3\rangle(3,2,1) = (2,1,3)$.

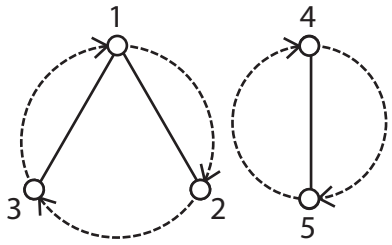


Figure 3.3: The digraph $\mathrm{dig}(\pi)$ of the permutation $\pi = \langle 1\ 2\ 3\rangle\langle 4\ 5\rangle$ and the graph $\mathrm{mgr}(\tau)$, for the transform $\tau = (\langle 1\ 2\rangle, \langle 1\ 3\rangle, \langle 4\ 5\rangle)$. Edges of $\mathrm{dig}(\pi)$ and $\mathrm{mgr}(\tau)$ are represented with dashed and solid lines, respectively.

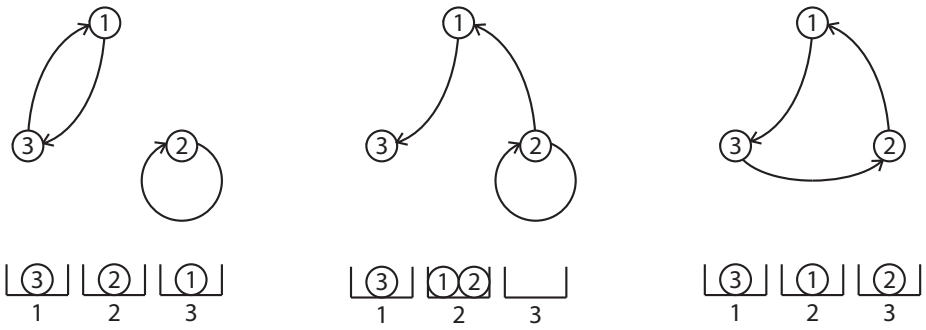


Figure 3.4: The transposition $\langle 2\ 3\rangle$ has an equivalent effect as first applying $(1, (3 \to 2))$ and then $(2, (2 \to 3))$. The first row shows the functional digraph of this operation and the second row shows its bins and balls representation.

In step j, for $j \in [k]$, the transposition $\tau_j = \langle x_j\ y_j\rangle$ is applied: the balls placed in bins x_j and y_j, moving along the edge $(x_j y_j)$, swap their locations. Note that the balls "move" solely along the edges of $\mathrm{mgr}\,(\tau)$. In this process, the ball $\pi(i)$ moves from bin i to bin $\pi(i)$. Thus there exists a path from i to $\pi(i)$ in $\mathrm{mgr}\,(\tau)$. □

A similar argument leads to the following corollary.

Corollary 3.3. *For permutations π and σ and a transform $\tau \in \mathbb{T}(\pi, \sigma)$, the following equivalent statements hold.*

- *There is a path from $\pi^{-1}(i)$ to $\sigma^{-1}(i)$ in $\mathrm{mgr}\,(\tau)$ for each $i \in [n]$.*
- *There is a path from i to $\sigma^{-1}(\pi(i))$ in $\mathrm{mgr}\,(\tau)$ for each $i \in [n]$.*
- *There is a path from i to $\pi^{-1}(\sigma(i))$ in $\mathrm{mgr}\,(\tau)$ for each $i \in [n]$.*

We define a generalization of the notion of a transposition where a can be moved independently of b. For example, let $a, b, c, d \in [n]$, and let $\pi(a) = c$ and $\pi(b) = d$. Let $\pi' = \pi\,(c, (a \to b))$, where $(c, (a \to b))$ denotes what we call a half-transposition, or *h-transposition* for short. This h-transposition moves c, the ball in bin a, to bin b, without moving ball a. We now have a mapping π' where $\pi'(b) = \{c, d\}$ and $\pi'(a) = \{\}$. Note that π' is no longer a function from $[n]$ to $[n]$ but rather a function from $[n]$ to the power set of $[n]$. For a given weight function φ, we assign weight $\frac{1}{2}\varphi_{\langle 1\ 2\rangle}$ to h-transpositions $(x, (a \to b))$ and $(x, (b \to a))$, independent of the value of x.

A transposition represents the product of a pair of h-transpositions, as in

$$\pi\,\langle a\ b\rangle = \pi\,(\pi(a), (a \to b))\,(\pi(b), (b \to a))\,.$$

Figure 3.4 illustrates this notion, using both the functional digraph and the balls and bins representation.

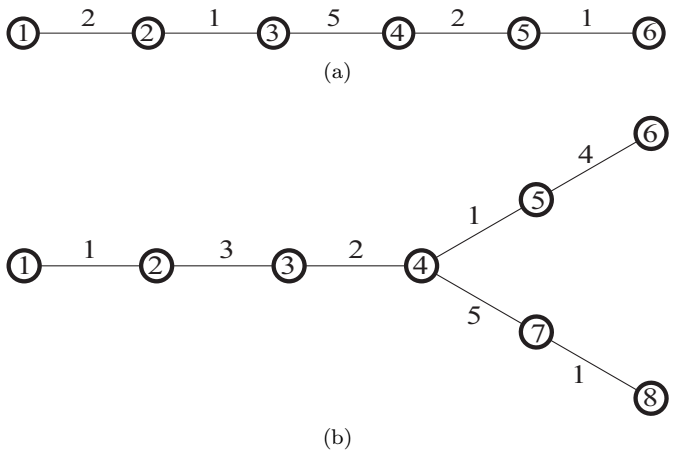


Figure 3.5: A defining path (a), which may correspond to a metric-path weight function or an extended-path weight function, and a defining tree (b), which may correspond to a metric-tree weight function or an extended-tree weight function.

An *h-transform* $\nu = (\nu_1, \ldots, \nu_{|\nu|})$ converting π to σ is a sequence of h-transpositions such that applying $\nu_1, \ldots, \nu_{|\nu|}$ to π, in that order, transforms π into σ. The weight of ν is defined as the sum of the weights of its constituent h-transpositions. Since transpositions can be modeled using h-transpositions, the minimum weight h-transform converting π to σ is at most the minimum weight transposition transform converting π to σ.

3.2.2 Weight Functions

A weight function φ is a *metric weight function* if it satisfies the triangle inequality in the sense that

$$\varphi_{\langle a\ b\rangle} \le \varphi_{\langle a\ c\rangle} + \varphi_{\langle c\ b\rangle}, \qquad a, b, c \in [n]. \tag{3.3}$$

If φ is a metric weight function, by repeated application of the triangle inequality, one has $\varphi_{\langle a\ b\rangle} \le \mathrm{wt}(p^*_\varphi(a, b))$. Since the edge (ab) is a path from a to b, we also have $\varphi_{\langle a\ b\rangle} \ge \mathrm{wt}(p^*_\varphi(a, b))$. Thus,

$$\varphi_{\langle a\ b\rangle} = \mathrm{wt}(p^*_\varphi(a, b)). \tag{3.4}$$

A weight function φ is a *metric-tree weight function* if there exists a weighted tree Θ over the vertex set $[n]$, such that for distinct $a, b \in [n]$, $\varphi_{\langle a\ b\rangle}$ is the sum of the weights of the edges on the unique path from a to b in Θ. If Θ is a path, i.e., if Θ is a linear graph, then φ is called a *metric-path weight function.*

Furthermore, a weight function φ' is an *extended-tree weight function* if there exists a weighted tree Θ over the vertex set $[n]$ such that for distinct $a, b \in [n]$,

$\varphi'_{\langle a\ b\rangle}$ equals the weight of the edge (ab) whenever a and b are neighbors in Θ, and $\varphi'_{\langle a\ b\rangle} = \infty$ otherwise. If Θ is a path, then φ' is called an *extended-path weight function*.

The tree or path corresponding to a weight function in the above definitions is termed the *defining tree or path* of the weight function. An example is given in Figure 3.5, where the numbers indexing the edges denote their weights.

Note that the weight functions used to define the weighted Kendall τ distance, which assign finite weights to adjacent transpositions, are extended path weight functions with the defining path $(1, \ldots, n)$.

3.3 Distance of a Transposition from Identity

Next, we find the distance between a transposition $\langle a\ b\rangle$, for $a, b \in [n]$, and the identity permutation e. Since the weighted transposition distance is left-invariant, the results of this section applies to any two permutations π and σ such that $\pi = \sigma\langle a\ b\rangle$ for some transposition $\langle a\ b\rangle$. We start by presenting two simple upper-bounds on the distance. The exact distance is characterized in Theorem 3.7.

For a weight function φ and a transposition $\langle a\ b\rangle$, one has

$$\mathsf{d}_\varphi(\langle a\ b\rangle, e) \le \varphi_{\langle a\ b\rangle}. \tag{3.5}$$

This simple upper-bound follows from the fact that $\langle a\ b\rangle \in \mathbb{T}(\langle a\ b\rangle, e)$. The following lemma presents a second upper-bound on the distance $\mathsf{d}_\varphi(\langle a\ b\rangle, e)$.

Lemma 3.4. *For a weight function φ and a transposition $\langle a\ b\rangle \in \mathbb{S}_n$,*

$$\mathsf{d}_\varphi(\langle a\ b\rangle, e) \le 2\mathrm{wt}(p^*_\varphi(a, b)).$$

Proof. Consider a path $p = (v_0, v_1, \ldots, v_{|p|})$ in K_φ, where $v_0 = a$ and $v_{|p|} = b$. We have

$$\begin{aligned}\langle a\ b\rangle = \langle v_0\ v_1\rangle\langle v_1\ v_2\rangle\cdots\left\langle v_{|p|-2}\ v_{|p|-1}\right\rangle\left\langle v_{|p|-1}\ v_{|p|}\right\rangle \\ \left\langle v_{|p|-2}\ v_{|p|-1}\right\rangle\cdots\langle v_1\ v_2\rangle\langle v_0\ v_1\rangle.\end{aligned}$$

From the left-invariance of d_φ, it follows that

$$\begin{aligned}\mathsf{d}_\varphi(\langle a\ b\rangle, e) &\le 2\sum_{i=1}^{|p|-1}\mathsf{d}_\varphi(\langle v_{i-1}\ v_i\rangle, e) - \mathsf{d}_\varphi\left(\left\langle v_{|p|-1}\ v_{|p|}\right\rangle, e\right) \\ &\le 2\sum_{i=1}^{|p|-1}\varphi_{\langle v_{i-1}\ v_i\rangle} - \varphi_{\left\langle v_{|p|-1}\ v_{|p|}\right\rangle} \\ &= 2\mathrm{wt}(p) - \varphi_{\left\langle v_{|p|-1}\ v_{|p|}\right\rangle} \\ &\le 2\mathrm{wt}(p).\end{aligned}$$

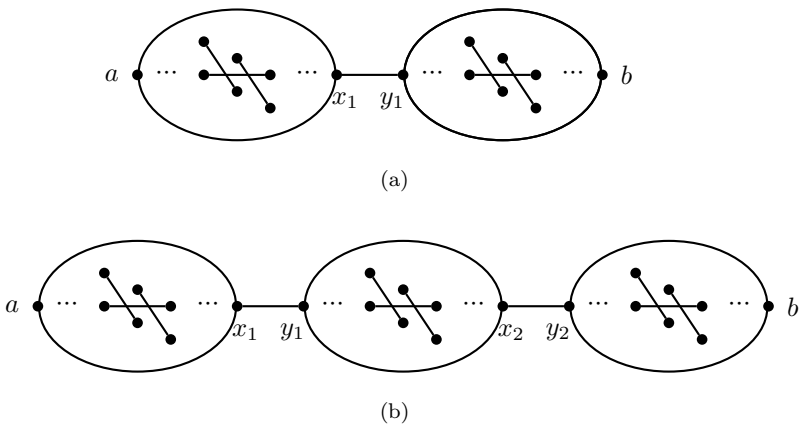


Figure 3.6: Illustrations for the proof of Lemma 3.5: (a) A graph corresponding to a transposition transform of $\langle a\ b\rangle$ with one cut edge only. The cut edge is (x_1y_1) and it separates component A containing a and component B containing b. (b) A graph corresponding to a transposition transform of $\langle a\ b\rangle$ with two cut edges. The cut edges are (x_1y_1), which separates component A containing a and component B containing y_1, and (x_2y_2), which separates components B and C containing b.

Since p is an arbitrary path from a to b in K_φ, we have

$$\mathsf{d}_\varphi\left(\langle a\ b\rangle, e\right) \leq 2\mathrm{wt}(p^*_\varphi(a,b))).$$

□

While the upper-bound of Lemma 3.4 suffices to prove many of our subsequent results, we also point out that one can compute the exact distance as shown in Theorem 3.7. Lemmas 3.5 and 3.6 are used in the proof of Theorem 3.7.

Lemma 3.5. *For a transposition $\langle a\ b\rangle$ and a sequence $\tau \in \mathbb{T}(\langle a\ b\rangle, e)$, the graph $\mathrm{mgr}(\tau)$ has at most one a,b-cut edge.*

Proof. Let $\mathcal{M} = \mathrm{mgr}(\tau)$. By Lemma 3.2, there is a path from a to b in $\mathcal{M}$. Suppose that $\tau = (\tau_1, \ldots, \tau_i, \ldots, \tau_k)$, and that $\tau_i = \langle x_1\ y_1\rangle$. Furthermore, suppose that (x_1y_1) is an a,b-cut edge as shown in Figure 3.6a. The graph $\mathcal{M} - (x_1y_1)$ has two components with vertex sets A and B, containing a and b, respectively. Since there exists a path from a to b, there also exists a path from a to x_1 that does not use the edge (x_1y_1). Thus, a and x_1 are in A. Similarly, b and y_1 are in B.

For $1 \leq j \leq k$, let $\pi_j = \pi_0\tau_1\cdots\tau_j$, where $\pi_0 = \langle a\ b\rangle$. Note that $\pi_0^{-1}(a) = b$ and $\pi_0^{-1}(b) = a$. For $i \in [n]\backslash\{a,b\}$, $\pi_0^{-1}(i) = i$. Recall that applying a transposition $\langle i\ j\rangle$ to a permutation swaps the successors of i and j. Since there is no transposition in π_{i-1} with endpoints in both A and B, we have $\pi_0^{-1}(a) \in B$ and

$\pi_0^{-1}(b) \in A$. Since (x_1y_1) is the only edge connecting A and B, we must have

$$\begin{aligned} \pi_{i-1}(x_1) &= b, & \pi_{i-1}(y_1) &= a, \\ \pi_i(x_1) &= a, & \pi_i(y_1) &= b. \end{aligned} \tag{3.6}$$

Also, for all $j \geq i$, we must have

$$\pi_j^{-1}(a) \in A, \qquad \pi_j^{-1}(b) \in B. \tag{3.7}$$

Alternatively, one can prove (3.6) and (3.7) by referring to the balls and bins model. Starting from the permutation $\langle a\ b\rangle$, in which bins a and b are respectively occupied by balls b and a, by applying the transpositions $\tau_1, \ldots, \tau_k$, ball a is moved to bin a and ball b is moved to bin b, while all other balls remain in their original bins. Since (x_1y_1) is the only edge connecting the components A and B, by applying $\tau_i = \langle x_1\ y_1\rangle$, the balls a and b are moved from one component to the other. This implies (3.6). Furthermore, after applying $\langle x_1\ y_1\rangle$, ball a remains in A and ball b remains in B, implying (3.7).

Now suppose there are at least two a,b-cut edges in $\mathcal{M}$. Let $\tau_i = \langle x_1\ y_1\rangle$ and $\tau_l = \langle x_2\ y_2\rangle$, with $l > i$. Suppose that (x_1y_1) and (x_2y_2) are two of the a,b-cut edges as shown in Figure 3.6b. Define A, B, and C to be the vertex sets of the components of $\mathcal{M} - (x_1y_1) - (x_2y_2)$ containing a, y_1, and y_2, respectively. By the same reasoning as above,

$$\pi_{l-1}(x_2) = b, \qquad \pi_{l-1}(y_2) = a.$$

However, this cannot be true, since for all $j \geq i$ we have $\pi_j^{-1}(a) \in A$ and thus $\pi_{l-1}^{-1}(a) \neq y_2 \in C$. This contradiction shows that $\mathcal{M}$ cannot contain more than one a,b-cut edge. □

Lemma 3.6. *For $a, b \in [n]$, among graphs with the same vertex set as that of K_φ and with at most one a,b-cut edge, there exists a graph of minimum weight that consists of two copies of a path from a to b with one edge deleted (removed).*

Proof. Let $\mathcal{M}$ denote the graph of minimum weight among graphs with the same vertex set as that of K_φ and with at most one a,b-cut edge. First, suppose that $\mathcal{M}$ has no a,b-cut edge. By deleting edges from $\mathcal{M}$, we may obtain a graph with one a,b-cut edge and of weight smaller than or equal to the weight of $\mathcal{M}$. Thus, we may assume that $\mathcal{M}$ has exactly one a,b-cut edge.

Suppose the a,b-cut edge in $\mathcal{M}$ is (xy) and that a and x are in the same component of $\mathcal{M} - (xy)$. There is no a,x-cut edge in $\mathcal{M}$, and thus, by Menger's theorem [41], there are two edge-disjoint paths from a to x. Delete the path with larger weight and instead add a copy of the path with smaller weight. Similarly, delete the path with larger weight from b to y and add a copy of the path with smaller weight. The graph obtained in this way has weight smaller than or equal to the weight of $\mathcal{M}$ and has one a,b-cut edge. Furthermore, it consists of two

copies of a path from a to b with one edge deleted. Namely, it has two copies of every edge in a path $(a, \dots, x, y, \dots, b)$ except for (xy), of which it has only one copy. □

Theorem 3.7. *For a weight function φ and a transposition $\langle a\ b\rangle \in \mathbb{S}_n$,*

$$\mathsf{d}_\varphi(\langle a\ b\rangle, e) = \min_{p=(v_0=a, v_1, \dots, v_{|p|}=b)} \left(2\mathrm{wt}(p) - \max_{0\le i<|p|} \varphi_{\langle v_i\ v_{i+1}\rangle}\right), \tag{3.8}$$

where the minimum is taken over all paths p from a to b.

Proof. Let d denote the right side of (3.8). By Lemmas 3.5 and 3.6, we have $d \le \mathsf{d}_\varphi(\langle a\ b\rangle, e)$. To prove that $d \ge \mathsf{d}_\varphi(\langle a\ b\rangle, e)$, let $p = (v_0, \dots, v_l)$ be a path of length l from $v_0 = a$ to $v_l = b$. For any $0 \le i < l$, we show that there is a sequence $\tau \in \mathbb{T}(\langle a\ b\rangle, e)$ such that

$$\mathrm{wt}(\tau) = 2\mathrm{wt}(p) - \varphi_{\langle v_i\ v_{i+1}\rangle}.$$

Namely, observe that

$$\begin{aligned}\tau = \big(&\langle v_0\ v_1\rangle, \langle v_1\ v_2\rangle, \dots, \langle v_{i-2}\ v_{i-1}\rangle, \langle v_{i-1}\ v_i\rangle,\\ &\langle v_{l-1}\ v_l\rangle, \langle v_{l-2}\ v_{l-i}\rangle, \dots, \langle v_{i+1}\ v_{i+2}\rangle, \langle v_{i+2}\ v_{i+3}\rangle,\\ &\langle v_i\ v_{i+1}\rangle,\\ &\langle v_{i-1}\ v_i\rangle, \langle v_{i-2}\ v_{i-1}\rangle, \dots, \langle v_1\ v_2\rangle, \langle v_0\ v_1\rangle,\\ &\langle v_{i+1}\ v_{i+2}\rangle, \langle v_{i+2}\ v_{i+3}\rangle, \dots, \langle v_{l-2}\ v_{l-1}\rangle, \langle v_{l-1}\ v_l\rangle\big).\end{aligned} \tag{3.9}$$

To see that indeed $\tau \in \mathbb{T}(\langle a\ b\rangle, e)$, observe that

$$\begin{aligned}\langle v_0\ v_1\rangle\langle v_1\ v_2\rangle\cdots\langle v_{i-2}\ v_{i-1}\rangle\langle v_{i-1}\ v_i\rangle &= \langle v_0\ v_1\ \cdots\ v_i\rangle,\\ \langle v_{l-1}\ v_l\rangle\langle v_{l-2}\ v_{l-i}\rangle\dots\langle v_{i+2}\ v_{i+3}\rangle\langle v_{i+1}\ v_{i+2}\rangle &= \langle v_l\ v_{l-1}\ \cdots\ v_{i+1}\rangle,\\ \langle v_{i-1}\ v_i\rangle\langle v_{i-2}\ v_{i-1}\rangle\dots\langle v_1\ v_2\rangle\langle v_0\ v_1\rangle &= \langle v_i\ v_{i-1}\ \cdots\ v_0\rangle,\\ \langle v_{i+1}\ v_{i+2}\rangle\langle v_{i+2}\ v_{i+3}\rangle\dots\langle v_{l-2}\ v_{l-1}\rangle\langle v_{l-1}\ v_l\rangle &= \langle v_{i+1}\ v_{i+2}\ \cdots\ v_l\rangle,\end{aligned} \tag{3.10}$$

and that

$$\langle a\ b\rangle = \langle v_0\ v_1\ \cdots\ v_i\rangle\langle v_l\ v_{l-1}\ \cdots\ v_{i+1}\rangle\langle v_i\ v_{i+1}\rangle\langle v_i\ v_{i-1}\ \cdots\ v_0\rangle\langle v_{i+1}\ v_{i+2}\ \cdots\ v_l\rangle.$$

By the definition of d_φ, and the fact that $\tau \in \mathbb{T}(\langle a\ b\rangle, e)$, we have

$$\mathrm{wt}(\tau) = 2\mathrm{wt}(p) - \varphi_{\langle v_i\ v_{i+1}\rangle} \ge \mathsf{d}_\varphi(\langle a\ b\rangle, e).$$

Since p is an arbitrary path from a to b, and since i is an arbitrary integer such that $0 \le i < |p|$, we have $d \ge \mathsf{d}_\varphi(\langle a\ b\rangle, e)$. □

In the next two subsections, we present two algorithms for computing the expression on the right side of (3.8). The first algorithm is based on writing

transpositions in terms of products of three transpositions with smaller total weights and the second algorithm is a modification of the well-known Bellman-Ford procedure [43] that searches the graph K_φ for a path that minimizes the right side of (3.8).

3.3.1 The Triple-Optimization Algorithm

In this subsection, we present an algorithm for finding the distance between a transposition $\langle a\ b\rangle \in \mathbb{S}_n$ and the identity permutation e for an arbitrary weight function φ. The unique transform in $\mathbb{T}(\langle a\ b\rangle, e)$ with length 1 is $\langle a\ b\rangle$. Since $\langle a\ b\rangle$ is odd, there are no transforms in $\mathbb{T}(\langle a\ b\rangle, e)$ with length 2. For $c \in [n]\backslash\{a, b\}$,

$$(\langle a\ c\rangle, \langle b\ c\rangle, \langle a\ c\rangle) \tag{3.11}$$

is a transform in $\mathbb{T}(\langle a\ b\rangle, e)$ of length 3.

If $\varphi_{\langle a\ b\rangle} > 2\varphi_{\langle a\ c\rangle} + \varphi_{\langle b\ c\rangle}$, then $(\langle a\ c\rangle, \langle b\ c\rangle, \langle a\ c\rangle)$ has a smaller weight than $\varphi_{\langle a\ b\rangle}$. Furthermore, one can substitute $\langle a\ c\rangle$ by, say, $(\langle c\ d\rangle, \langle a\ d\rangle, \langle c\ d\rangle)$ to obtain the transform

$$(\langle c\ d\rangle, \langle a\ d\rangle, \langle c\ d\rangle, \langle b\ c\rangle, \langle c\ d\rangle, \langle a\ d\rangle, \langle c\ d\rangle)$$

with yet smaller weight if $\varphi_{\langle a\ c\rangle} > 2\varphi_{\langle c\ d\rangle} + \varphi_{\langle a\ d\rangle}$. The same procedure may be repeated until it is no longer possible to reduce the weight any further. The resulting transform is a *triple-optimized transform* for the transposition $\langle a\ b\rangle$. The weight of such a transform is the *triple-optimized weight* of $\langle a\ b\rangle$, and is denoted by $\varphi^*_{\langle a\ b\rangle}$. While there may be many distinct triple-optimized transforms for a transposition $\langle a\ b\rangle$, as we will see, they all must have the same weight, namely, $\mathsf{d}_\varphi(\langle a\ b\rangle, e)$. Note that, by definition, $\varphi^*_{\langle a\ b\rangle} \le 2\varphi^*_{\langle a\ x\rangle} + \varphi^*_{\langle b\ x\rangle}$, for any $x \neq a, b$.

We develop an algorithm – Alg. 2 – that finds triple-optimized transforms for all transpositions. Alg. 2 performs a simple search on the ordered set of transpositions to check if their product, of the form given in (3.11), yields a transform of lower weight for some transposition. It then updates the weights of transpositions and repeats. The fact that Alg. 2 produces the triple-optimized weights $\varphi^*_{\langle a\ b\rangle}$, for $\langle a\ b\rangle \in \mathbb{S}_n$, is shown in Lemma 3.8. In Theorem 3.10, we prove that triple-optimized transforms are optimum, that is, $\varphi^*_{\langle a\ b\rangle} = \mathsf{d}_\varphi(\langle a\ b\rangle, e)$ for $\langle a\ b\rangle \in \mathbb{S}_n$.

The input to Alg. 2 is an ordered list Ω of transpositions and their weights. Each item of Ω corresponds to one transposition and is of the form $\left[\langle a\ b\rangle | \varphi_{\langle a\ b\rangle}\right]$. The jth item of Ω is denoted by $\Omega(j)$ and the transposition corresponding to $\Omega(j)$ is denoted by $T\Omega(j)$. Sorting of Ω means reordering its items so that transpositions appear in increasing order of their weights. The output of the algorithm is a list with the same format, but with triple-optimized weights for transpositions.

Algorithm 2 OPTIMIZE-TRANSPOSITION-WEIGHTS(Ω)

```
1: Input: Ω (the list of transpositions and their weights)
2: Sort Ω
3: for i ← 2, ..., |Ω| do
4:     ⟨a_1 b_1⟩ ← TΩ(i)
5:     φ_1 ← φ_⟨a_1 b_1⟩
6:     for j ← 1, ..., i − 1 do
7:         ⟨a_2 b_2⟩ ← TΩ(j)
8:         φ_2 ← φ_⟨a_2 b_2⟩
9:         if {a_1 b_1} ∩ {a_2 b_2} ≠ {} then
10:            a_com ← {a_1, b_1} ∩ {a_2, b_2}
11:            {a_3, b_3} ← {a_1, a_2, b_1, b_2} \ {a_com}
12:            if φ_1 + 2φ_2 < φ_⟨a_3 b_3⟩ then
13:                φ_⟨a_3 b_3⟩ ← φ_1 + 2φ_2
14:                update φ_⟨a_3 b_3⟩ in Ω
15:    Sort Ω
```

Lemma 3.8. *Alg. 2 outputs the triple-optimized weight φ^* for all transpositions in $\mathbb{S}_n$.*

Proof. Let Ω_i be the list Ω at the beginning of iteration i, obtained immediately before executing line 4 of Alg. 2. Also, let $\Omega_i(j,\dots,k)$ denote the items $j, j+1,\dots,k$ of Ω_i and let $T\Omega_i(j,\dots,k)$ denote the transpositions corresponding to these items.

We prove by induction that $T\Omega_i(1,\dots,i)$ are triple-optimized, that the items in $\Omega_i(1,\dots,i)$ do not change in subsequent iterations of the algorithm, and that $\Omega_i(i+1,\dots,|\Omega|)$ are triple-optimized with respect to $\Omega_i(1,\dots,i-1)$, i.e., any triple-optimized transform of a transposition $\langle a\ b\rangle \in T\Omega_i(i+1,\dots,|\Omega|)$ in terms of $T\Omega_i(1,\dots,i-1)$ has weight at least as large as the weight of $\langle a\ b\rangle$ in $\Omega_i(i+1,\dots,|\Omega|)$.

The claim is obviously true for $i = 2$.

Assume that the claim holds for i. Note that, by the induction hypotheses, $\Omega_i(1,\dots,i) = \Omega_{i+1}(1,\dots,i)$ and $T\Omega_i(i+1,\dots,|\Omega|) = T\Omega_{i+1}(i+1,\dots,|\Omega|)$. Let $\langle a_1\ b_1\rangle = T\Omega_i(i)$, and consider a transposition $s \in T\Omega_{i+1}(i+1,\dots,|\Omega|)$. By the induction hypotheses, s is already triple-optimized with respect to $\Omega_{i+1}(1,\dots,i-1)$. Thus, the weight of s may be reduced using transpositions $T\Omega_{i+1}(1,\dots,i)$ only if one can write s as $\langle a_2\ b_2\rangle\langle a_1\ b_1\rangle\langle a_2\ b_2\rangle$, where $\langle a_2\ b_2\rangle \in T\Omega_{i+1}(1,\dots,i-1)$. These are precisely the transforms considered in the ith iteration, and thus $\Omega_{i+1}(i+1,\dots,|\Omega|)$ is triple-optimized with respect to $\Omega_{i+1}(1,\dots,i)$.

Furthermore, $T\Omega_{i+1}(i{+}1)$ has the minimum weight among $T\Omega_{i+1}(i{+}1,\dots,|\Omega|)$, and thus its weight cannot be further reduced. Hence, the weights of transpositions $T\Omega_{i+1}(1,\dots,i{+}1)$ are triple-optimized and do not change in the subsequent iterations of the algorithm.

□

Example 3.9. The left-most list in (3.12) represents the input Ω to the algo-

rithm, with transpositions in increasing order of their weights. This is equal to Ω_2. The two lists that follow represent updates of Ω produced by Alg. 2.

In the first iteration, the algorithm considers the transposition $\langle 1\ 3\rangle$, for $i = 2$, and the transposition $\langle 3\ 4\rangle$, for $j = 1$. Using these transpositions we may write $\langle 3\ 4\rangle\langle 1\ 3\rangle\langle 3\ 4\rangle = \langle 1\ 4\rangle$. The initial weight of $\langle 1\ 4\rangle$ is 12 which exceeds $2\varphi_{\langle 3\ 4\rangle} + \varphi_{\langle 1\ 3\rangle} = 8$ and thus Ω_3 is obtained, i.e., the second list in (3.12).

Next, for $i = 3$ and $j = 1$, the algorithm considers $\langle 2\ 4\rangle$ and $\langle 3\ 4\rangle$. Since $\langle 3\ 4\rangle\langle 2\ 4\rangle\langle 3\ 4\rangle = \langle 2\ 3\rangle$, we update the weight of $\langle 2\ 3\rangle$ from 23 to 11 as shown in the third list Ω_4 in (3.12). Additional iterations of the algorithm introduce no further changes in the weights.

$$\left[\begin{array}{c|c} \langle 3\ 4\rangle & 2 \\ \langle 1\ 3\rangle & 4 \\ \langle 2\ 4\rangle & 7 \\ \langle 1\ 4\rangle & 12 \\ \langle 1\ 2\rangle & 15 \\ \langle 2\ 3\rangle & 23 \end{array}\right] \to \left[\begin{array}{c|c} \langle 3\ 4\rangle & 2 \\ \langle 1\ 3\rangle & 4 \\ \langle 2\ 4\rangle & 7 \\ \langle \mathit{1}\ \mathit{4}\rangle & \mathit{8} \\ \langle 1\ 2\rangle & 15 \\ \langle 2\ 3\rangle & 23 \end{array}\right] \to \left[\begin{array}{c|c} \langle 3\ 4\rangle & 2 \\ \langle 1\ 3\rangle & 4 \\ \langle 2\ 4\rangle & 7 \\ \langle 1\ 4\rangle & 8 \\ \langle \mathit{2}\ \mathit{3}\rangle & \mathit{11} \\ \langle 1\ 2\rangle & 15 \end{array}\right] \tag{3.12}$$

□

Next, we state a theorem pertaining to the optimality of Alg. 2.

Theorem 3.10. *For $\langle a\ b\rangle \in \mathbb{S}_n$, we have $\varphi^*_{\langle a\ b\rangle} = \mathsf{d}_\varphi(\langle a\ b\rangle, e)$.*

Proof. Consider the transposition $\langle a\ b\rangle$, and let τ be a sequence of transpositions that transforms $\langle a\ b\rangle$ into e with $\mathrm{wt}(\tau) = \mathsf{d}_\varphi(\langle a\ b\rangle, e)$. Furthermore, let $\mathcal{M} = \mathrm{mgr}(\tau)$. Let $\mathcal{M}'$ be a graph with minimum weight among those consisting of two copies of a path from a to b minus one edge. By Lemmas 3.5 and 3.6, the weight of $\mathcal{M}'$ is less than or equal to the weight of $\mathcal{M}$. We show below that the weight obtained from Alg. 2 is, in turn, less than or equal to the weight of $\mathcal{M}'$. By optimality of $\mathcal{M}$, this proves the theorem.

Suppose that $\mathcal{M}'$ consists of two copies of the path

$$(a, x_1, x_2, \ldots, x_r, x, y, y_1, y_2, \ldots, y_s, b),$$

for some integers r and s, minus one copy of (xy). Thus, we have

$$\begin{aligned} C_1 &:= 2\varphi^*_{\langle a\ x_1\rangle} + \cdots + 2\varphi^*_{\langle x_r\ x\rangle} + \varphi^*_{\langle x\ y\rangle} \\ &\geq 2\varphi^*_{\langle a\ x_1\rangle} + \cdots + 2\varphi^*_{\langle x_{r-1}\ x_r\rangle} + \varphi^*_{\langle x_r\ y\rangle} \\ &\geq 2\varphi^*_{\langle a\ x_1\rangle} + \cdots + 2\varphi^*_{\langle x_{r-2}\ x_{r-1}\rangle} + \varphi^*_{\langle x_{r-1}\ y\rangle} \\ &\geq \cdots \\ &\geq \varphi^*_{\langle a\ y\rangle}. \end{aligned} \tag{3.13}$$

The weight of $\mathcal{M}'$ is $C_1 + 2\varphi^*_{\langle y\ y_1\rangle} + 2\varphi^*_{\langle y_1\ y_2\rangle} + \cdots + 2\varphi^*_{\langle y_s\ b\rangle}$, and thus we have

$$\begin{aligned} C_1 + 2\varphi^*_{\langle y\ y_1\rangle} + 2\varphi^*_{\langle y_1\ y_2\rangle} + \cdots + 2\varphi^*_{\langle y_s\ b\rangle} &\geq \varphi^*_{\langle a\ y\rangle} + 2\varphi^*_{\langle y\ y_1\rangle} + \cdots + 2\varphi^*_{\langle y_s\ b\rangle} \\ &\geq \varphi^*_{\langle a\ y_1\rangle} + 2\varphi^*_{\langle y_1\ y_2\rangle} + \cdots + 2\varphi^*_{\langle y_s\ b\rangle} \\ &\geq \varphi^*_{\langle a\ y_2\rangle} + 2\varphi^*_{\langle y_2\ y_3\rangle} + \cdots + 2\varphi^*_{\langle y_s\ b\rangle} \\ &\geq \cdots \\ &\geq \varphi^*_{\langle a\ y_s\rangle} + 2\varphi^*_{\langle y_s\ b\rangle} \\ &\geq \varphi^*_{\langle a\ b\rangle}, \end{aligned} \tag{3.14}$$

implying that $\varphi^*_{\langle a\ b\rangle} \leq \mathsf{d}_\varphi(\langle a\ b\rangle, e)$. Hence, $\varphi^*_{\langle a\ b\rangle} = \mathsf{d}_\varphi(\langle a\ b\rangle, e)$. □

Observe that if the weight function satisfies

$$\varphi_{\langle b\ c\rangle} + 2\varphi_{\langle a\ c\rangle} \geq \varphi_{\langle a\ b\rangle}, \text{ for distinct } a, b, c \in [n], \tag{3.15}$$

then

$$\mathsf{d}_\varphi(\langle a\ b\rangle, e) = \varphi^*_{\langle a\ b\rangle} = \varphi_{\langle a\ b\rangle}. \tag{3.16}$$

In particular, (3.16) holds for metric weight functions.

Computational Complexity: For each index i, the number of operations performed is $O(|\Omega|)$. Thus, the complexity of the algorithm is $O(|\Omega|^2)$. Since $|\Omega|$ is at most equal to the number of transpositions, we have $|\Omega| = \binom{n}{2}$. Hence, the complexity of Alg. 2 equals $O(n^4)$.

3.3.2 The Bellman-Ford Algorithm

We describe an algorithm for finding the path that solves

$$\min_{p \in P(a,b)} \left(2\mathrm{wt}(p) - \max_{(xy)\in p} \varphi_{\langle x\ y\rangle} \right),$$

for $a, b \in [n]$, where $P(a,b)$ is the set of all paths from a to b in K_φ. Recall from Theorem 3.7 that the weight of this path equals $\mathsf{d}_\varphi(\langle a\ b\rangle, e)$. The algorithm presented here is a variant of the Bellman-Ford algorithm. The Bellman-Ford algorithm is described in detail in [43].

Recall that for each path $p = (v_1, \ldots, v_{k+1})$ in K_φ, the standard weight of the path is

$$\mathrm{wt}\,(p) = \sum_{i=1}^{k} \varphi_{\langle v_i\ v_{i+1}\rangle}. \tag{3.17}$$

Define the *transposition path weight* of p

$$\bar{\varphi}\,(p) := 2\mathrm{wt}\,(p) - \max_i \varphi_{\langle v_i\ v_{i+1}\rangle} \tag{3.18}$$

to be the weight of the graph consisting of two copies of p minus one copy of its heaviest edge. For $s, u \in [n], s \neq u$, let the path that minimizes the transposition

path weight, among all paths from s to u, be denoted by $\hat{p}(s,u)$. The goal is to find $\hat{p}(s,u)$, for all $s,u \in [n], s \neq u$.

Before describing our algorithm, we briefly review the standard Single-Source Bellman-Ford shortest path algorithm, and its relaxation technique.

Given a fixed source s, for each vertex $v \neq s$, the algorithm maintains an upper-bound on the minimum weight of a path from s to v, denoted by $\mathbb{D}(v)$.

"Relaxing" an edge (uv) means testing that the upper-bounds $\mathbb{D}(u)$ and $\mathbb{D}(v)$ satisfy the conditions

$$\begin{aligned} \mathbb{D}(u) &\leq \mathbb{D}(v) + w, \\ \mathbb{D}(v) &\leq \mathbb{D}(u) + w, \end{aligned} \tag{3.19}$$

where w denotes the weight of the edge (uv). If the above conditions are not satisfied, then one of the two upper-bounds can be improved, since one can reach u by passing through v, and vice versa.

In our algorithm, we maintain the upper-bound for two types of weights. The source s is an arbitrary vertex in K_φ. We use $\mathbb{D}_1(v)$ to denote the upper-bound on the minimum transposition path weight of a path from s to v, and we use $\mathbb{D}_2(v)$ to denote the upper-bound on twice the minimum weight of a path from s to v. From the definitions of these weights, the relaxation inequalities become

$$\begin{aligned} \mathbb{D}_2(u) &\leq 2w + \mathbb{D}_2(v), \\ \mathbb{D}_2(v) &\leq 2w + \mathbb{D}_2(u), \\ \mathbb{D}_1(u) &\leq \min\{w + \mathbb{D}_2(v), 2w + \mathbb{D}_1(v)\}, \\ \mathbb{D}_1(v) &\leq \min\{w + \mathbb{D}_2(u), 2w + \mathbb{D}_1(u)\}. \end{aligned} \tag{3.20}$$

The relaxation algorithm for these inequalities, Alg. 3, is straightforward to implement.

To describe the properties of the output of our Bellman-Ford algorithm, we briefly comment on a simple property of the algorithm, termed the path-relaxation property.

Suppose $p = (v_1, \ldots, v_{k+1})$ is the shortest path in terms of (3.17) (resp. (3.18)) from $s = v_1$ to $u = v_{k+1}$. After relaxing the edges $(v_1v_2), (v_2v_3), \ldots, (v_kv_{k+1})$, in that given order, the upper-bound $\mathbb{D}_2(u)$ (resp. $\mathbb{D}_1$) equals twice the minimum weight (resp. the minimum transposition path weight) of a path from s to u. Note that the property still holds even if the relaxations of the edges $(v_1v_2), (v_2v_3), \ldots, (v_kv_{k+1})$ are interleaved by relaxations of some other edges.

In the algorithm below, we use $pred_i(v)$ to denote the predecessor of node v used for tracking the updates of the weight $\mathbb{D}_i(v)$, $i = 1, 2$, and $(u, i), i = 1, 2$, to indicate from which of the two weights, minimized over in (3.20), u originated. Note that this notion of predecessor is not to be confused with the predecessor of an element in the functional digraph of a permutation.

The modified Bellman-Ford algorithm, Alg. 4, performs $n-1$ rounds of relax-

Algorithm 3 RELAX(u, v)

```
1: w ← φ(u, v)
2: if D_2(v) > D_2(u) + 2w then
3:     D_2(v) ← D_2(u) + 2w
4:     pred_2(v) ← (u, 2)
5: if D_2(u) > D_2(v) + 2w then
6:     D_2(u) ← D_2(v) + 2w
7:     pred_2(u) ← (v, 2)
8: if D_1(v) > D_2(u) + w then
9:     D_1(v) ← D_2(u) + w
10:    pred_1(v) ← (u, 2)
11: if D_1(u) > D_2(v) + w then
12:    D_1(u) ← D_2(v) + w
13:    pred(u, 1) ← (v, 2)
14: if D_1(v) > D_1(u) + 2w then
15:    D_1(v) ← D_1(u) + 2w
16:    pred_1(v) ← (u, 1)
17: if D_1(u) > D_1(v) + 2w then
18:    D_1(u) ← D_1(v) + 2w
19:    pred_1(u) ← (v, 1)
```

ation on the edges of the graph K_φ. Lemma 3.11 proves the correctness of the algorithm.

An example of the steps of Alg. 4 is given in Figure 3.7. As a result of the relaxation of edges (ab) and (ac), neighbors of a have finite weights, as shown in Figure 3.7a. Edge (bd) is relaxed next, as seen in Figure 3.7b. Then, the relaxation of edge (cd) reduces $\mathbb{D}_1(d)$ from 12 to 10. Continuing with the algorithm, we obtain the final result in Figure 3.7f. Note that in this example, the result obtained after the first pass is the final result. In general, however, the final weights may be obtained only after all $n-1$ passes are performed.

Lemma 3.11. *Given n, a weight function φ, and a source s, after the execution of Alg. 4, one has $\mathbb{D}_1(u) = \mathsf{d}_\varphi(\langle s\ u\rangle, e)$ and $\mathbb{D}_2(u) = 2\mathrm{wt}(p^*(s,u))$.*

Proof. Let $\hat{p}(s,u) = (v_1, v_2, \ldots, v_{k+1})$ be the path that minimizes $\bar{\varphi}(p)$ among all paths p from $v_1 = s$ to $v_{k+1} = u$. Since any path p has at most n vertices, we have $k \leq n-1$. The algorithm makes $n-1$ passes and in each pass relaxes all edges of the graph. Thus, there exists a subsequence of relaxations that relax $(v_1v_2), (v_2v_3), \ldots, (v_kv_{k+1})$, in that order. The proof for the claim regarding $\mathbb{D}_1(u)$ follows by invoking the path-relaxation property and the fact that $\mathsf{d}_\varphi(\langle s\ u\rangle, e) = \bar{\varphi}(\hat{p}(s,u))$. The proof for the claim regarding $\mathbb{D}_2(u)$ is similar. □

Algorithm 4 SINGLE-SOURCE BELLMAN-FORD(s)

```
Input: vertex s
Output: p̂(s,u) for 1 ≤ u ≤ n
for u ← 1···n do
    𝔻₁(u) ← ∞
    𝔻₂(u) ← ∞
𝔻₁(s) ← 0
𝔻₂(s) ← 0
for i ← 1···n − 1 do
    for each edge (uv) ∈ E(K_φ) do
        RELAX(u, v)
for u ← 1···n do
    initialize path at u
    backtrack min weight alg to recover path to s
    output p̂(s,u)
```

3.4 Distance between General Permutations

In the previous section, we presented algorithms for finding the weighted transposition distance between two permutations that differ by a single transposition. In this section, we study the distance between arbitrary permutations. For general weight functions and permutations, an algorithm for computing the distance is not known. We present approximation methods for the general case as well as exact solutions for special weight functions.

Lemma 3.12. *For a weight function φ and for $\pi, \sigma \in \mathbb{S}_n$,*

$$\mathsf{d}_\varphi(\pi, \sigma) \le 2D_\varphi(\pi, \sigma).$$

If φ is a metric weight function, the bound may be improved to

$$\mathsf{d}_\varphi(\pi, \sigma) \le D_\varphi(\pi, \sigma).$$

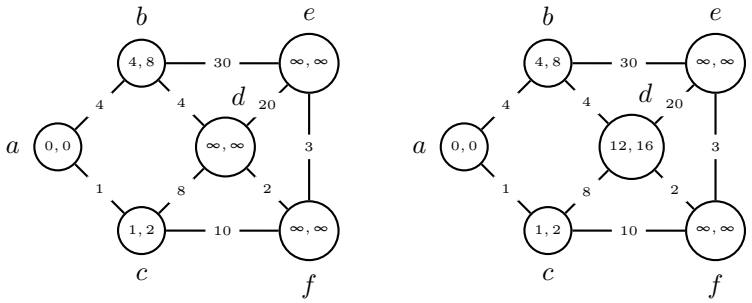
Proof. To prove the first claim, it suffices to show that $\mathsf{d}_\varphi(\pi, e) \le 2D_\varphi(\pi, e)$, since both d_φ and D_φ are left-invariant.

Let $\{c_1, c_2, \ldots, c_k\}$ be the cycle decomposition of π. We have, from the triangle inequality and the left-invariance property of d_φ, that

$$\mathsf{d}_\varphi(\pi, e) \le \sum_{i=1}^{k} \mathsf{d}_\varphi(c_i, e),$$

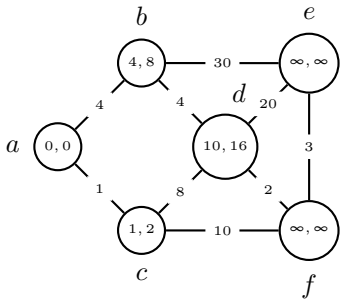
and, from the definition of D_φ,

$$D_\varphi(\pi, e) = \sum_{i=1}^{k} D_\varphi(c_i, e).$$

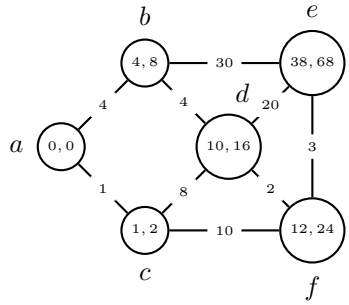


(a) Relaxation of (ab) and (ac)

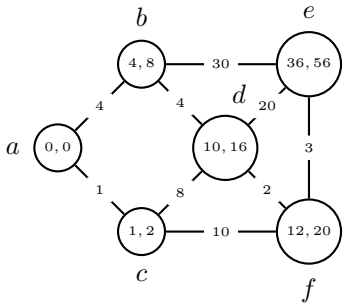
(b) Relaxation of (bd)



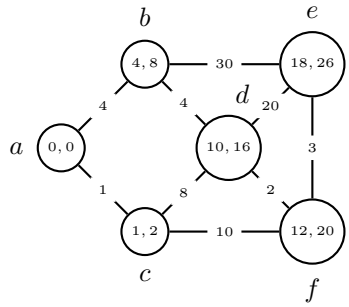
(c) Relaxation of (cd)



(d) Relaxation of (be) and (cf)



(e) Relaxation of (de) and (df)



(f) Relaxation of (fe).

Figure 3.7: Single-Pair Bellman-Ford algorithm on a 6-vertex graph. The weights $(\mathbb{D}_1(u), \mathbb{D}_2(u))$, are shown inside each vertex. Edges that are not drawn have weight ∞.

Hence, we only need to prove

$$\mathsf{d}_\varphi(c, e) \le 2D_\varphi(c, e) \tag{3.21}$$

for a single cycle $c = \langle a_1\ a_2\ \cdots\ a_{|c|} \rangle$, where $|c|$ is the length of c.

Since c may be written as

$$c = \langle a_1\ a_2 \rangle \langle a_2\ a_3 \rangle \cdots \langle a_{|c|-1}\ a_{|c|} \rangle,$$

we have

$$\begin{aligned}\mathsf{d}_\varphi(c,e) &\le \sum_{i=1}^{|c|-1} \varphi_{\langle a_i\ a_{i+1}\rangle} \\ &\overset{(a)}{\le} \sum_{i=1}^{|c|-1} 2\mathrm{wt}(p^*_\varphi(a_i,a_{i+1})) \\ &\le \sum_{i=1}^{|c|} 2\mathrm{wt}(p^*_\varphi(a_i,c(a_i))) \\ &\le 2D_\varphi(c,e)\end{aligned}$$

where (a) follows from Lemma 3.4.

The proof of the second claim is similar, except that we write

$$\begin{aligned}\mathsf{d}_\varphi(c,e) &\le \sum_{i=1}^{|c|-1} \varphi_{\langle a_i\ a_{i+1}\rangle} \\ &\overset{(b)}{=} \sum_{i=1}^{|c|-1} \mathrm{wt}(p^*_\varphi(a_i,a_{i+1})) \\ &\le \sum_{i=1}^{|c|} \mathrm{wt}(p^*_\varphi(a_i,c(a_i))) \\ &\le 2D_\varphi(c,e)\end{aligned}$$

where (b) follows from (3.4).

□

The next lemma provides a lower-bound for d_φ in terms of D_φ.

Lemma 3.13. *For $\pi,\sigma \in \mathbb{S}_n$,*

$$\mathsf{d}_\varphi(\pi,\sigma) \ge \frac{1}{2} D_\varphi(\pi,\sigma).$$

Proof. Since d_φ and D_φ are both left-invariant, it suffices to show that

$$\mathsf{d}_\varphi(\pi,e) \ge \frac{1}{2} D_\varphi(\pi,e).$$

Let $(\tau_1,\ldots,\tau_l)$, with $\tau_j = \langle a_j\ b_j\rangle$, be a minimum weight transform of π into e, so that $\mathsf{d}_\varphi(\pi,e) = \sum_{i=1}^{l} \varphi_{\langle a_j\ b_j\rangle}$. Furthermore, define $\pi_j = \pi\tau_1\cdots\tau_j$, $0 \le j \le l$. Then,

$$\begin{aligned}D_\varphi(\pi_{j-1},e) - D_\varphi(\pi_j,e) &\le 2\mathrm{wt}\left(p^*_\varphi(a_j,b_j)\right) \\ &\le 2\varphi_{\langle a_j\ b_j\rangle}, \end{aligned} \tag{3.22}$$

where the first inequality follows from considering the maximum possible decrease of the value of D_φ induced by one transposition, while the second inequality follows from the definition of p^*_φ. By summing up the terms in (3.22) over $0 \le j \le l$, and thus obtaining a telescoping inequality of the form $D_\varphi(\pi,e) \le$

$2\sum_{i=1}^{l} \varphi_{\langle a_j\ b_j\rangle} = 2\mathsf{d}_\varphi(\pi, e)$, we arrive at the desired result. □

An alternative proof of Lemma 3.13 can be given in terms of h-transpositions. Recall from §3.2.1 that

$$\pi\left(\pi(a), (a \to b)\right)\left(\pi(b), (b \to a)\right) = \pi\,\langle a\ b\rangle$$

and that the weight of each h-transposition is half the weight of the corresponding transposition. Any transform can be modeled as an h-transform with the same weight by breaking each transposition into two h-transpositions. Thus, the minimum weight of a transform converting π to e is at least as large as the minimum weight of an h-transform converting π to e. The minimum weight h-transform uses the shortest path $p^*_\varphi\left(\pi^{-1}(i), i\right)$ between $\pi^{-1}(i)$ and i for each $i \in [n]$:

$$(i, (v_0 \to v_1))\,(i, (v_1 \to v_2)) \cdots (i, (v_{k-1} \to v_k))\,,$$

where $p^*_\varphi\left(\pi^{-1}(i), i\right) = (v_0, v_1, , \ldots, v_k)$ is the shortest path from $v_0 = \pi^{-1}(i)$ to $v_k = i$ in K_φ. This h-transform has weight

$$\frac{1}{2}\sum_i \mathrm{wt}\left(p^*_\varphi\left(\pi^{-1}(i), i\right)\right) = \frac{1}{2} D_\varphi(\pi, e),$$

and this completes the proof.

From the previous two lemmas, we have the following theorem.

Theorem 3.14. *For $\pi, \sigma \in \mathbb{S}_n$ and an arbitrary nonnegative weight function φ, we have*

$$\frac{1}{2} D_\varphi(\pi, \sigma) \le \mathsf{d}_\varphi(\pi, \sigma) \le 2 D_\varphi(\pi, \sigma).$$

In addition, if φ is a metric weight function, then

$$\frac{1}{2} D_\varphi(\pi, \sigma) \le \mathsf{d}_\varphi(\pi, \sigma) \le D_\varphi(\pi, \sigma).$$

For special classes of the weight function φ, the bounds in Theorem 3.14 may be improved further, or exact solutions may be obtained, as described in the next subsection.

3.4.1 Distance for Metric-Path and Metric-Tree Weights

This section is devoted to metric-path and metric-tree weight functions. The exact distance for metric-path weight functions can be obtained in polynomial time as shown in Lemma 3.15. We also show that for some permutations, the exact distance for metric-tree weight functions can also be computed efficiently.

Since metric-path and metric-tree weight functions are metric weight functions, the second part of Theorem 3.14 applies to these cases. Additionally,

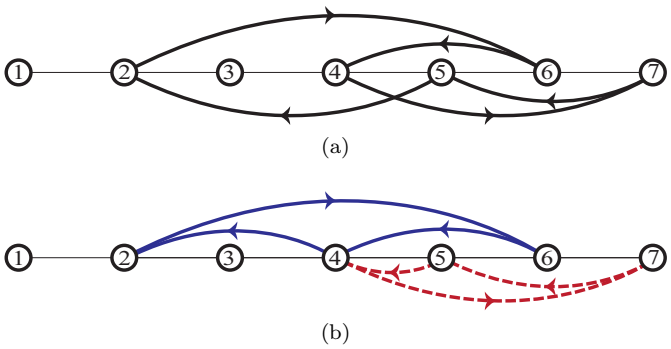


Figure 3.8: The cycle $\langle 2\ 6\ 4\ 7\ 5\rangle$ in Figure (a) is decomposed into two cycles, $\langle 2\ 6\ 4\rangle$ and $\langle 4\ 7\ 5\rangle$, depicted in Figure (b). Note that $\langle 2\ 6\ 4\ 7\ 5\rangle = \langle 2\ 6\ 4\rangle\langle 4\ 7\ 5\rangle$.

(3.16) implies that

$$\mathsf{d}_\varphi(\langle a\ b\rangle, e) = \varphi_{\langle a\ b\rangle} \tag{3.23}$$

for all transpositions $\langle a\ b\rangle$ and a metric-path or a metric-tree weight function φ.

Lemma 3.15. *For a metric-path weight function φ and for $\pi, \sigma \in \mathbb{S}_n$,*

$$\mathsf{d}_\varphi(\pi, \sigma) = \frac{1}{2} D_\varphi(\pi, \sigma).$$

Proof. From Lemma 3.13, we have that $\mathsf{d}_\varphi(\pi,\sigma) \geq \frac{1}{2} D_\varphi(\pi,\sigma)$. It remains to show that $\mathsf{d}_\varphi(\pi,\sigma) \leq \frac{1}{2} D_\varphi(\pi,\sigma)$. Since d_φ and D_φ are both left-invariant, it suffices to prove that $\mathsf{d}_\varphi(\pi, e) \leq \frac{1}{2} D_\varphi(\pi, e)$.

Let $\{c_1, c_2, \ldots, c_k\}$ be the cycle decomposition of π. Similar to the proof of Lemma 3.12, it suffices to show that

$$\mathsf{d}_\varphi(c, e) \leq \frac{1}{2} D_\varphi(c, e) \tag{3.24}$$

for any cycle $c = \langle a_1 \cdots a_{|c|}\rangle$.

The proof is by induction. For $|c| = 2$, (3.24) holds since from (3.23) we have

$$\mathsf{d}_\varphi(\langle a_1\ a_2\rangle, e) = \varphi_{\langle a_1\ a_2\rangle} = \mathrm{wt}\left(p^*_\varphi(a_1, a_2)\right) = \frac{1}{2} D_\varphi(\langle a_1\ a_2\rangle, e).$$

Assume that (3.24) holds for $2 \leq |c| < l$. We show that it also holds for $|c| = l$. We use Figure 3.8 as an illustration. In all figures in this section, undirected edges describe the defining tree, while directed edges describe the cycle at hand.

Without loss of generality, assume that the defining path of φ, Θ, equals $(1, 2, \ldots, n)$. Furthermore, assume that $a_1 = \min\{i : i \in c\}$; if this is not the case, we can rewrite c by cyclically shifting its elements. Let $a_t = \min\{i : i \in c, i \neq a_1\}$ be the "closest" element to a_1 in Θ (that is, the closest element to a_1 in c). For

example, in Figure 3.8, one has $c = \langle 2\ 6\ 4\ 7\ 5\rangle$, $a_1 = 2$ and $a_t = 4$. We have

$$\begin{aligned} c &= \langle a_1\ a_2 \cdots a_t \cdots a_l\rangle \\ &= \langle a_1 \cdots a_t\rangle \langle a_t \cdots a_l\rangle \end{aligned} \tag{3.25}$$

and thus

$$\begin{aligned} \mathsf{d}_\varphi(c,e) &\le \mathsf{d}_\varphi(\langle a_1 \cdots a_t\rangle, e) + \mathsf{d}_\varphi(\langle a_t \cdots a_l\rangle, e) \\ &\le \frac{1}{2}\sum_{i=1}^{t-1} \mathrm{wt}\left(p^*_\varphi(a_i,a_{i+1})\right) + \mathrm{wt}\left(p^*_\varphi(a_t,a_1)\right) \\ &\quad + \frac{1}{2}\sum_{i=t}^{l-1} \mathrm{wt}\left(p^*_\varphi(a_i,a_{i+1})\right) + \mathrm{wt}\left(p^*_\varphi(a_l,a_t)\right) \\ &= \frac{1}{2}\sum_{i=1}^{l} \mathrm{wt}\left(p^*_\varphi(a_i,c(a_i))\right) \\ &= \frac{1}{2} D_\varphi(c,e), \end{aligned}$$

where the second inequality follows from the induction hypothesis, while the first equality follows from the fact that

$$\mathrm{wt}\left(p^*_\varphi(a_t,a_1)\right) + \mathrm{wt}\left(p^*_\varphi(a_l,a_t)\right) = \mathrm{wt}\left(p^*_\varphi(a_l,a_1)\right).$$

□

Remark 3.16. The inductive proof of the above lemma provides us with a transform converting a permutation π to e with weight $\frac{1}{2}D_\varphi(\pi,e)$ for a metric-path weight function φ. Using (3.25) recursively, each cycle c of π can be written as a product $\tau_1 \cdots \tau_k$ of transpositions. The transform $(\tau_1,\ldots,\tau_k)$ converts c to e. The concatenation of transforms of cycles of π is a transform converting π to e and has weight $\frac{1}{2}D_\varphi(\pi,e)$.

The approach described in the proof of Lemma 3.15 can also be applied to the problem of finding the weighted transposition distance when the weight function is a metric-tree weight function and *each of the cycles of the permutation* consists of elements that lie on some path in the defining tree. An example of such a permutation and such a weight function is shown in Figure 3.9. Note that in this example, a cycle consisting of elements $3, 5, 7$ would not correspond to a path.

In such a case, for each cycle c of π we can use the path in the defining tree that contains the elements of c to show that

$$\mathsf{d}_\varphi(c,e) = \frac{1}{2} D_\varphi(c,e). \tag{3.26}$$

For example the cycle $\langle 1\ 4\ 6\rangle$ lies on the path $(1,2,3,4,5,6)$ and the cycle $\langle 5\ 8\rangle$

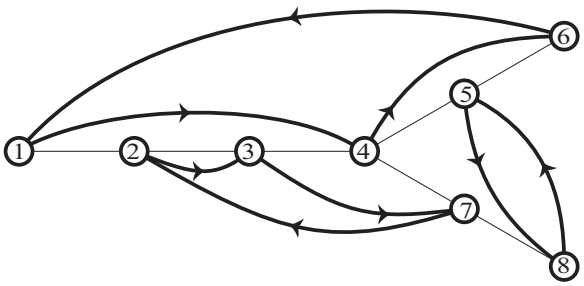


Figure 3.9: If each of the cycles of a permutation lies on a path, the method of Lemma 3.15 can be used to find the weighted transposition distance.

lies on the path $(5, 4, 7, 8)$. Since (3.26) holds for each cycle c of π, we have

$$\mathsf{d}_\varphi(\pi, e) = \frac{1}{2} D_\varphi(\pi, e).$$

A similar scenario in which essentially the same argument as that of the proof of Lemma 3.15 can be used is as follows: the defining tree has one vertex with degree three and no vertices with degree larger than three (i.e., a tree with a Y shape), and for each cycle of π, there are two branches of the tree that do not contain two consecutive elements of c. It can then be shown that each such cycle can be decomposed into cycles that lie on paths in the defining tree, reducing the problem to the previously described one. An example is shown in Figure 3.10.

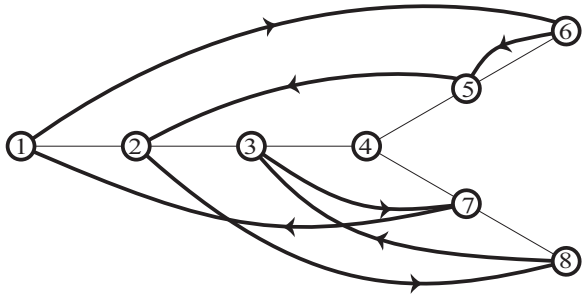
One may argue that the results of Lemma 3.15 and its extension to metric-trees have limited application, as they require that both the defining tree and the permutations used in the computation be of special form. In particular, one may require that a given ranking π is such that there are *no edges* between two different branches of Θ in the cycle graph of π. We show next that under certain conditions the probability of such permutations goes to zero as $n \to \infty$, by lower bounding the number P_n of permutations with the given constraint.

Let the set of vertices in the ith branch of a Y shaped defining tree Θ, $i = 1, 2, 3$, be denoted by B_i and let b_i denote the number of vertices in B_i. Clearly, $b_1 + b_2 + b_3 + 1 = n$.

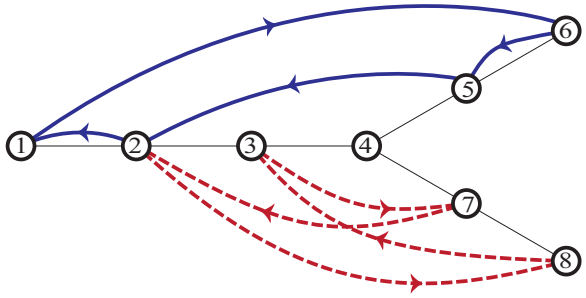
Assume, without loss of generality, that the numbering of the branches is such that $b_1 \geq b_2 \geq b_3$. As an illustration, in Figure 3.10 we have

$$\begin{aligned} B_1 &= \{1, 2, 3\}, \quad & b_1 &= 3, \\ B_2 &= \{5, 6\}, & b_2 &= 2, \\ B_3 &= \{7, 8\}, & b_3 &= 2. \end{aligned}$$

The quantity P_n is greater than or equal to the number of permutations π whose functional digraph does not contain an edge between B_2 and B_3, and



(a)



(b)

Figure 3.10: The cycle $\langle 1\ 6\ 5\ 2\ 8\ 3\ 7\rangle$ in Figure (a) is decomposed into two cycles, $\langle 1\ 6\ 5\ 2\rangle$ and $\langle 2\ 8\ 3\ 7\rangle$, as shown in Figure (b). Note that $\langle 1\ 6\ 5\ 2\ 8\ 3\ 7\rangle = \langle 1\ 6\ 5\ 2\rangle\langle 2\ 8\ 3\ 7\rangle$.

this quantity is, in turn, greater than or equal to the number of permutations π such that $\pi(j) \notin B_2 \cup B_3$ for $j \in B_2 \cup B_3$. The number of permutation with the latter property equals $\binom{b_1+1}{b_2+b_3}(b_2+b_3)!(b_1+1)!$. Hence,

$$P_n \geq \frac{((b_1+1)!)^2}{(b_1+1-b_2-b_3)!},$$

and thus

$$\frac{P_n}{n!} \geq \frac{\prod_{j=n+1-2b_2-2b_3}^{n-b_2-b_3} j}{\prod_{j=n+1-b_2-b_3}^{n} j}.$$

In particular, if $b_2 = b_3 = 1$, we have

$$\frac{P_n}{n!} \geq \frac{(n-3)(n-2)}{(n-1)n} = 1 - \frac{4}{n} + O(n^{-2})$$

and more generally, if $b_2 + b_3 = o(n)$, then

$$\frac{P_n}{n!} \geq \frac{(n+o(n))^{b_2+b_3}}{(n+o(n))^{b_2+b_3}} \sim 1,$$

or equivalently, $P_n \sim n!$.

Hence, if $b_2 + b_3 = o(n)$, the distance $\mathsf{d}_\varphi(\pi, e)$ of a randomly chosen permu-

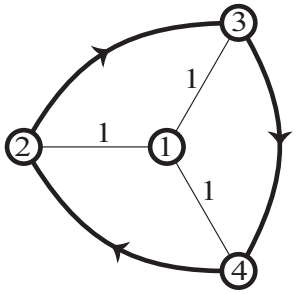


Figure 3.11: For the above metric-tree weight function and $\pi = \langle 2\ 3\ 4 \rangle$, the equality of Lemma 3.15 does not hold.

tation π from the identity equals $D_\varphi(\pi, e)/2$ with probability approaching 1 as $n \to \infty$.

It is worth noting that for metric-tree weight functions, the equality of Lemma 3.15 is not, in general, satisfied. To prove this claim, consider the metric-tree weight function φ of Figure 3.11, where, for $a, b \in [4], a < b$,

$$\varphi_{\langle a\ b \rangle} = \begin{cases} 1, & \text{if } a = 1, \\ \infty, & \text{if } a \neq 1. \end{cases}$$

It can be shown that for the permutation $\pi = \langle 2\ 3\ 4 \rangle$, $\mathsf{d}_\varphi(\pi, e) = 4$, while $\frac{1}{2} D_\varphi(\pi, e) = 3$.

3.4.2 Distance for Extended-Path Weight Functions

The following lemma provides a 2-approximation for transposition distances based on extended-path weight functions. As the weighted Kendall distance is a special case of the weighted transposition distance with extended-path weight functions, the lemma also implies Lemma 2.13.

Lemma 3.17. *For an extended-path weight function φ and for $\pi, \sigma \in \mathbb{S}_n$,*

$$\frac{1}{2} D_\varphi(\pi, \sigma) \leq \mathsf{d}_\varphi(\pi, \sigma) \leq D_\varphi(\pi, \sigma).$$

Proof. The lower-bound follows from Lemma 3.13. To prove the upper-bound, consider a metric-path weight function φ', with the same defining path Θ as φ, such that

$$\varphi'_{\langle a\ b \rangle} = 2\varphi_{\langle a\ b \rangle}$$

for any pair a, b adjacent in Θ. From Lemma 3.4, it follows that for distinct $c, d \in [n]$,

$$\mathsf{d}_\varphi(\langle c\ d \rangle, e) \leq 2\mathrm{wt}(p^*_\varphi(c, d)) = \mathrm{wt}(p^*_{\varphi'}(c, d)) = \mathsf{d}_{\varphi'}(\langle c\ d \rangle, e).$$

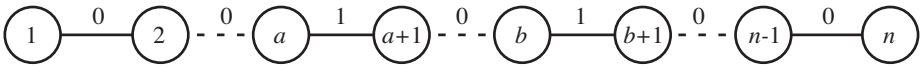


Figure 3.12: The graph of a weight function with two non-zero weights $\varphi_{\langle a\ a+1\rangle}$ and $\varphi_{\langle b\ b+1\rangle}$, with $\varphi_{\langle a\ a+1\rangle} = \varphi_{\langle b\ b+1\rangle} = 1$.

Hence,

$$\mathsf{d}_{\varphi}(\pi,\sigma) \le \mathsf{d}_{\varphi'}(\pi,\sigma) = \frac{1}{2} D_{\varphi'}(\pi,\sigma) = D_{\varphi}(\pi,\sigma),$$

which proves the claimed result. □

Remark 3.18. Based on Remark 3.16, the proof of the above lemma provides us with a transform of weight at most $D_{\varphi}(\pi, e)$ converting a permutation π to e, for an extended-path weight function φ.

3.4.3 Weight Functions with Two Identical Non-zero Weights

The goal is to find the weighted Kendall distance $\mathsf{d}_{\varphi}(\pi, e)$, for $\pi \in \mathbb{S}_n$, with the weight function of (2.12),

$$\varphi_{\langle i\ i+1\rangle} = \begin{cases} 1, & i \in \{a, b\} \\ 0, & \text{else.} \end{cases}$$

For this purpose, let $R_1 = \{1, \ldots, a\}$, $R_2 = \{a+1, \ldots, b\}$, and $R_3 = \{b+1, \ldots, n\}$, and define

$$N_{ij}^{\pi} = |\{k \in R_j : \pi^{-1}(k) \in R_i\}|, \quad i, j \in \{1, 2, 3\}.$$

That is, N_{ij}^{π} is the number of elements whose ranks in π belong to the set R_i and whose ranks in e belong to the set R_j. A sequence of transpositions that transforms π into e moves the N_{ij}^{π} elements of $\{k \in R_j : \pi^{-1}(k) \in R_i\}$ from R_i to R_j. Furthermore, note that any transposition that swaps two elements with ranks in the same region $R_i, i \in [3]$, has weight zero, while for any transposition τ_l that swaps an element ranked in R_1 with an element ranked in R_2 or swaps an element ranked in R_2 with an element ranked in R_3, we have $\mathsf{d}_{\varphi}(\tau_l, e) = 1$.

It is straightforward to see that $\sum_j N_{ij}^{\pi} = \sum_j N_{ji}^{\pi}$. In particular, $N_{12}^{\pi} + N_{13}^{\pi} = N_{21}^{\pi} + N_{31}^{\pi}$ and $N_{31}^{\pi} + N_{32}^{\pi} = N_{13}^{\pi} + N_{23}^{\pi}$.

We show next that

$$\mathsf{d}_{\varphi}(\pi, e) = \begin{cases} 2N_{13}^{\pi} + N_{12}^{\pi} + N_{23}^{\pi}, & \text{if } N_{21}^{\pi} \ge 1 \text{ or } N_{23}^{\pi} \ge 1, \\ 2N_{13}^{\pi} + 1, & \text{if } N_{21}^{\pi} = N_{23}^{\pi} = 0. \end{cases}$$

Note that, from Lemma 2.13, we have

$$\mathsf{d}_{\varphi}(\pi, e) \ge \frac{1}{2} D_{\varphi}(\pi, e) = 2N_{13}^{\pi} + N_{12}^{\pi} + N_{23}^{\pi}. \tag{3.27}$$

Suppose that $N_{21}^{\pi} \geq 1$ or $N_{23}^{\pi} \geq 1$. We find a transposition τ_l, with $\mathsf{d}_{\varphi}(\tau_l, e) = 1$, such that $\pi' = \pi\tau_l$ satisfies $D_{\varphi}(\pi', e) = D_{\varphi}(\pi, e) - 2$, and at least one of the following conditions holds:

$$\begin{cases} \quad N_{21}^{\pi'} \geq 1, \\ \text{or} \\ \quad N_{23}^{\pi'} \geq 1, \\ \text{or} \\ \quad \pi' = e. \end{cases} \tag{3.28}$$

Applying the same argument repeatedly, and using the triangle inequality, proves that $\mathsf{d}_{\varphi}(\pi, e) \leq \frac{1}{2} D_{\varphi}(\pi, e)$ if $N_{21}^{\pi} \geq 1$ or $N_{23}^{\pi} \geq 1$. This, along with (3.27), shows that $\mathsf{d}_{\varphi}(\pi, e) = \frac{1}{2} D_{\varphi}(\pi, e)$ if $N_{21}^{\pi} \geq 1$ or $N_{23}^{\pi} \geq 1$.

First, suppose that $N_{21}^{\pi} \geq 1$ and $N_{23}^{\pi} \geq 1$. It then follows that $N_{12}^{\pi} \geq 1$ or $N_{32}^{\pi} \geq 1$. Without loss of generality, assume that $N_{12}^{\pi} \geq 1$. Then τ_l can be chosen such that $N_{12}^{\pi'} = N_{12}^{\pi} - 1$ and $N_{21}^{\pi'} = N_{21}^{\pi} - 1$. We have $D_{\varphi}(\pi', e) = D_{\varphi}(\pi, e) - 2$, and since $N_{23}^{\pi} \geq 1$, condition (3.28) holds.

Next, suppose $N_{21}^{\pi} \geq 1$ and $N_{23}^{\pi} = 0$. If $N_{13}^{\pi} \geq 1$, choose τ_l such that

$$\begin{aligned} N_{21}^{\pi'} &= N_{21}^{\pi} - 1, \\ N_{23}^{\pi'} &= 1, \\ N_{13}^{\pi'} &= N_{13}^{\pi} - 1, \end{aligned}$$

where $\pi' = \pi\tau_l$. Since $N_{23}^{\pi'} = 1$, condition (3.28) is satisfied. If $N_{13}^{\pi} = 0$, then $N_{31}^{\pi} = N_{32}^{\pi} = 0$, and thus $N_{12}^{\pi} = N_{21}^{\pi} \geq 1$. In this case, we choose τ_l such that $N_{21}^{\pi'} = N_{12}^{\pi'} = N_{12}^{\pi} - 1$. As a result, we have either $N_{21}^{\pi'} \geq 1$ or $\pi' = e$. Hence, condition (3.28) is satisfied once again. Note that in both cases, for $N_{13}^{\pi} = 0$ as well as for $N_{13}^{\pi} \geq 1$, we have $D_{\varphi}(\pi', e) = D_{\varphi}(\pi, e) - 2$.

The proof for the case $N_{23}^{\pi} \geq 1$ and $N_{21}^{\pi} = 0$ follows along similar lines.

If $N_{21}^{\pi} = N_{23}^{\pi} = 0$, it can be verified by inspection that for every transposition τ_l with $\mathsf{d}_{\varphi}(\tau_l, e) = 1$, we have $D_{\varphi}(\pi\tau_l, e) \geq D_{\varphi}(\pi, e)$. Hence, the inequality in (3.27) cannot be satisfied with equality, which implies that $\mathsf{d}_{\varphi}(\pi, e) \geq 2N_{13}^{\pi} + 1$. Choose a transposition τ_l with $\mathsf{d}_{\varphi}(\tau_l, e) = 1$ such that

$$\begin{aligned} N_{13}^{\pi'} &= N_{13}^{\pi} - 1, \\ N_{12}^{\pi'} &= 1, \\ N_{23}^{\pi'} &= 1, \end{aligned}$$

where $\pi' = \pi\tau_l$. We have

$$\mathsf{d}_{\varphi}(\pi, e) \leq \mathsf{d}_{\varphi}(\tau_l, e) + \mathsf{d}_{\varphi}(\pi', e) = 1 + 2N_{13}^{\pi}.$$

This, along with $\mathsf{d}_{\varphi}(\pi, e) \geq 2N_{13}^{\pi} + 1$, completes the proof.

3.5 Minimum Weight, Minimum Length Transposition Transforms

In Theorem 3.14, we established D_φ as an approximation for d_φ for any weight function φ. In this section, we introduce a second approximation, which is at least as good as D_φ and in some cases is strictly better.

Consider two permutations π and σ of $[n]$ and a weight function φ. As mentioned before, no efficient algorithm is known for finding the minimum weight transposition transform converting π to σ. But, as we show in this section, if one limits the search space to transforms with minimum length, the minimum weight transform can be computed efficiently. This limited search space leads to the minimum length transform distance between π and σ, denoted by $\mathsf{L}_\varphi(\pi,\sigma)$, and defined as

$$\mathsf{L}_\varphi(\pi,\sigma) = \arg\min_{\tau\in\mathbb{M}(\pi,\sigma)} \sum_{i=1}^{|\tau|} \varphi^*_{\tau_i}$$

where $\mathbb{M}(\pi,\sigma)$ is the set of minimum length transposition transforms (MLTs) converting π to σ, where transpositions are multiplied on the right. Note that in the definition above, we have used φ^* and not φ. The reason for this is that for every transposition $\langle a\ b\rangle$, we have $\varphi^*_{\langle a\ b\rangle} = \mathsf{d}_\varphi(\langle a\ b\rangle, e) \le \varphi_{\langle a\ b\rangle}$. Thus, using φ^* in the definition leads to a better approximation of the weighted transposition distance. Throughout this section, all weights are computed using φ^* rather than φ.

The distance L_φ is left-invariant. To see this, observe that $\mathbb{M}(\pi,\sigma) = \mathbb{M}(\omega\pi,\omega,\sigma)$ since $\pi\tau_1\cdots\tau_{|\tau|} = \sigma\tau_1\cdots\tau_{|\tau|}$ if and only if $\omega\pi\tau_1\cdots\tau_{|\tau|} = \omega\sigma\tau_1\cdots\tau_{|\tau|}$, where $\omega\in\mathbb{S}_n$ and τ_i are transpositions. For this reason, it suffices to consider $\mathsf{L}_\varphi(\pi,e)$ for $\pi\in\mathbb{S}_n$. In this section, by a *transform of* π, we mean a transform that converts π to the identity e.

The length of the MLT converting π to e equals the Cayley distance: if π has ℓ cycles, then the length of an MLT converting π to e is $n-\ell$. Each transposition in an MLT of π breaks a cycle of length more than 1 into two cycles of shorter lengths. For example, consider $\pi = (3,1,2,5,4) = \langle 1\ 3\ 2\rangle\langle 4\ 5\rangle$. The transform $\tau = (\langle 1\ 2\rangle, \langle 2\ 3\rangle, \langle 4\ 5\rangle)$ in an MLT of length 3 of π.

Suppose π has the cycle decomposition $\{\sigma_1,\ldots,\sigma_\ell\}$ and that $\tau^{(i)}$, for $i\in[\ell]$, is an MLT of the cycle σ_i. Concatenation of the MLTs of the cycles σ_i yields an MLT of π. That is,

$$\left(\tau_1^{(1)},\ldots,\tau_{j_1}^{(1)},\ldots,\tau_1^{(\ell)},\ldots,\tau_{j_\ell}^{(\ell)}\right),$$

where j_i is the length of $\tau^{(i)}$, is an MLT of π. The converse is also true: every MLT of π can be obtained by concatenating the MLTs of its cycles and reordering the transpositions. Hence,

$$\mathsf{L}_\varphi(\pi,e) = \sum_{i=1}^{\ell} \mathsf{L}_\varphi(\sigma_i,e).$$

Therefore, to be able to compute L_φ it is sufficient to be able to compute $\mathsf{L}_\varphi(\sigma, e)$ for all cycles σ. In this section, for clarity of presentation, and without loss of generality, we assume $\sigma = \langle 1 \cdots k \rangle$.

Note that even for a single cycle σ, the weighted transposition distance $\mathsf{d}_\varphi(\sigma, e)$, need not be equal to $\mathsf{L}_\varphi(\sigma, e)$, as illustrated by the following example [61].

Example 3.19. Consider the cycle $\sigma = \langle 1 \cdots 5 \rangle$ with $\varphi_{\langle i\ j \rangle} = 3$, for $|i - j| = 1$, and $\varphi_{\langle i\ j \rangle} = 1$ otherwise, where the indices are taken modulo the length of the cycle, which in this case equals five.

It is easy to verify that the transform $(\langle 1\ 4 \rangle, \langle 1\ 3 \rangle, \langle 3\ 5 \rangle, \langle 2\ 4 \rangle, \langle 1\ 4 \rangle, \langle 1\ 3 \rangle)$ is a minimum weight transform of σ with weight six, i.e., $\mathsf{d}_\varphi(\sigma, e) = 6$. However, as can be computed with the methods described in this section, the weight of a minimum weight MLT is eight, i.e., $\mathsf{L}_\varphi(\sigma, e) = 8$. One such MLT is $\langle \langle 4\ 5 \rangle, \langle 1\ 3 \rangle, \langle 2\ 3 \rangle, \langle 1\ 4 \rangle \rangle$. We also have $D_\varphi(\sigma, e) = 10$ which is larger than $\mathsf{L}_\varphi(\sigma, e)$. □

3.5.1 Computing Minimum Weight MLTs

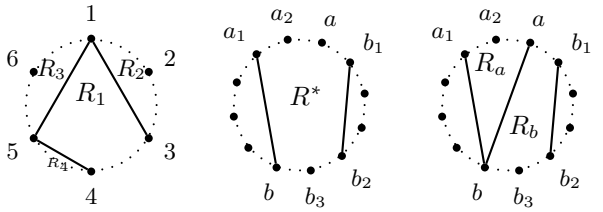
Recall that $\mathrm{dig}(\sigma)$ is drawn with its vertices on a circle. We prove in Lemma 3.20 that for an MLT τ of σ, $\mathrm{mgr}(\tau)$ is a tree such that $\mathrm{mgr}(\tau) \cup \mathrm{dig}(\sigma)$ is planar. Lemma 3.21 states that for every spanning tree T such that $\mathrm{dig}(\sigma) \cup T$ is planar, there is an MLT τ such that $T = \mathrm{mgr}(\tau)$. Hence, the search for minimum weight MLT can be performed over the space of spanning trees T with $\mathrm{dig}(\sigma) \cup T$ planar. This leads us to Alg. 5. The correctness of Alg. 5 is proved in Lemma 3.23.

The following definitions will be used in the proof of Lemma 3.20. Let R be the region enclosed by edges of $\mathrm{dig}(\sigma)$. Also, let T be a tree whose vertex set is a subset of $\{1, \ldots, k\}$ such that $\mathrm{dig}(\sigma) \cup T$ is planar. Since T is a tree with edges contained in R, the edges of T divide R into smaller regions, with the vertices and edges forming the boundaries included in all neighboring regions. The vertices $\{1, \ldots, k\}$ can be divided into *corner vertices* (lying at the intersection of at least two regions) and *inner vertices* (belonging only to one region). In Figure 3.13a, $\mathrm{dig}(\sigma)$ with vertices $\{1, 2, 3, 4, 5, 6\}$ is partitioned by T into four regions, R_1, R_2, R_3 and R_4. In R_2, vertices 1 and 3 are corner vertices, while vertex 2 is an inner vertex.

Lemma 3.20. *For a cycle σ and a transform $\tau \in \mathbb{M}(\sigma, e)$, the graph $\mathrm{mgr}(\tau)$ is a tree such that $\mathrm{mgr}(\tau) \cup \mathrm{dig}(\sigma)$ is planar.*

Proof. For each $i \in [k]$, by Corollary 3.3, there is a path from i to $\sigma(i)$ in $\mathrm{mgr}(\tau)$, so $\mathrm{mgr}(\tau)$ is connected with k vertices. Furthermore, $\mathrm{mgr}(\tau)$ has $k-1$ edges, since $|\tau| = k - 1$. Hence, $\mathrm{mgr}(\tau)$ is a tree.

For $i \in [k-1]$, let $\tau_1^i = (\tau_1, \ldots, \tau_i)$. The proof proceeds by showing that for all $1 \le i \le k-1$, the following two claims are true:



(a) T (b) $\sigma\tau_1\cdots\tau_i$ (c) $\sigma\tau_1\cdots\tau_{i+1}$

Figure 3.13: (a) A tree T divides R into four subregions. (b,c) A region is divided into two regions by transposition $\langle a\ b\rangle$. See proof of Lemma 3.20.

(I) The graph $\mathrm{dig}(\sigma) \cup \mathrm{mgr}\left(\tau_1^i\right)$ is planar.

(II) Each cycle of $\sigma\tau_1\cdots\tau_i$ corresponds to a subregion R of $\mathrm{dig}(\sigma) \cup \mathrm{mgr}\left(\tau_1^i\right)$. The cycle corresponding to R contains all of its inner vertices and some of its corner vertices but no other vertex.

Both claims (I) and (II) are obvious for $i = 1$. We show that if (I) and (II) are true for i, then they are also true for $i+1$.

Suppose $\tau_{i+1} = \langle a\ b\rangle$. Clearly, $\sigma\tau_1\cdots\tau_{i+1}$ has one more cycle than $\sigma\tau_1\cdots\tau_i$. By the induction hypothesis, $\mathrm{dig}(\sigma) \cup \mathrm{mgr}\left(\tau_1^i\right)$ is planar and partitioned into a set of subregions. Note that a and b are in the same cycle of $\sigma\tau_1\cdots\tau_i$, so they are inner or corner vertices of some subregion R^* of $\mathrm{dig}(\sigma) \cup \mathrm{mgr}\left(\tau_1^i\right)$. The edge (ab) divides R^* into two subregions, R_a and R_b (without crossing any edge in $\mathrm{dig}(\sigma) \cup \mathrm{mgr}\left(\tau_1^i\right)$). This proves (I).

Let the cycle of the permutation $\sigma\tau_1\cdots\tau_i$ that corresponds to R^* be given by $\mu = \langle a_1 \cdots a_l\ a\ b_1 \cdots b_{l'}\ b\rangle$, as seen in Figure 3.13b. Note that $\mu\langle a\ b\rangle = \langle a\ a_1 \cdots a_l\rangle\langle b\ b_1 \cdots b_{l'}\rangle$. Now the cycles $\langle a\ a_1 \cdots a_l\rangle$ and $\langle b\ b_1 \cdots b_{l'}\rangle$ in $\sigma\tau_1\cdots\tau_{i+1}$ correspond to subregions R_a and R_b, respectively, as seen in Figure 3.13c. This proves claim (II), since the cycle corresponding to each subregion contains all of its inner vertices and some of its corner vertices but no other vertex. □

For related ideas regarding permutation transforms and graphical structures, the interested reader is referred to [62].

The following lemma establishes a partial converse to the previous lemma.

Lemma 3.21. *For a cycle $\sigma = \langle 1 \cdots k\rangle$ and a spanning tree T over the vertices $\{1,\ldots,k\}$, if $\mathrm{dig}(\sigma) \cup T$ is planar, then there exists at least one MLT $\tau \in \mathbb{M}(\sigma, e)$ such that $T = \mathrm{mgr}(\tau)$.*

Proof. We prove the lemma by recursively constructing an MLT τ converting σ to e such that $T = \mathrm{mgr}(\tau)$. If $k = 2$, then T has exactly one edge and τ is the transposition corresponding to that edge. For $k > 2$, some vertex has degree larger than 1. Without loss of generality, assume that $\deg(1) > 1$. Let

$$r = \max\{u : (1u) \in T\}.$$

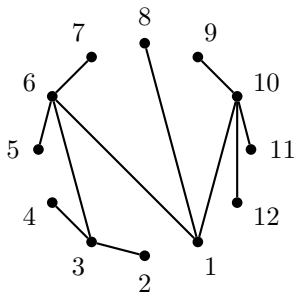


Figure 3.14: Example illustrating the proof of Lemma 3.21: $r = 10, s = 8$.

Since T is a tree, $T - (1r)$ has two components. These two components have vertex sets $\{1, \ldots, s\}$ and $\{s+1, \ldots, k\}$, for some s. It is easy to see that

$$\langle 1 \cdots k \rangle = \langle s+1 \cdots k\ 1 \rangle \langle 1 \cdots s \rangle . \tag{3.29}$$

Let

$$\begin{aligned} T' &= T[\{1, \ldots, s\}], \\ T'' &= T[\{s+1, \ldots, k, 1\}]. \end{aligned}$$

Note that T' and T'' have fewer than k vertices. Furthermore, $T' \cup \mathrm{dig}(\langle 1 \cdots s \rangle)$ and $T'' \cup \mathrm{dig}(\langle s+1 \cdots k\ 1 \rangle)$ are planar. Thus, by the induction hypothesis, $\langle 1 \cdots s \rangle$ and $\langle s+1 \cdots k\ 1 \rangle$ have MLTs τ' and τ'' converting them to the identity, of lengths $s-1$ and $k-s$, respectively. By (3.29), the transform

$$(\tau'_1, \ldots, \tau'_{s-1}, \tau''_1, \ldots, \tau''_{k-s})$$

is an MLT converting σ to the identity. Furthermore, $T = \mathrm{mgr}(\tau)$. □

Example 3.22. In Figure 3.14, we have $r = 10$ and $s = 8$. The cycle $\langle 1 \cdots 12 \rangle$ can be written as a product of two shorter cycles using (3.29),

$$\langle 1 \cdots 12 \rangle = \langle 9\ 10\ 11\ 12\ 1 \rangle \langle 1\ 2 \cdots 8 \rangle .$$

Now, each of these cycles is written as a product of two shorter cycles in a similar manner, for example,

$$\begin{aligned} \langle 9\ 10\ 11\ 12\ 1 \rangle &= \langle 9\ 10 \rangle \langle 10\ 11\ 12\ 1 \rangle , \\ \langle 1 \cdots 8 \rangle &= \langle 8\ 1 \rangle \langle 1 \cdots 7 \rangle . \end{aligned}$$

Algorithm 5 MIN-WEIGHT-MLT

Input: Optimized transposition weight function Φ^* where $\Phi^*_{i,j} = \varphi^*_{\langle i\ j\rangle}$ (Output of Alg. 2)
$C(i,j) \leftarrow \infty$ for $i,j \in [k]$
$C(i,i) \leftarrow 0$ for $i \in [k]$
$C(i,i+1) \leftarrow \varphi^*_{\langle i\ i+1\rangle}$ for $i \in [k]$
for $l = 2 \cdots k-1$ **do**
  for $i = 1 \cdots k-l$ **do**
    $j \leftarrow i+l$
    for $i \le s < r \le j$ **do**
      $A \leftarrow C(i,s) + C(s+1,r) + C(r,j) + \varphi^*_{\langle i\ r\rangle}$
      if $A < C(i,j)$ **then**
        $C(i,j) \leftarrow A$

□

Since any MLT in $\mathtt{M}(\sigma, e)$ of a cycle σ can be represented by a tree that is planar on the circle, the search for the minimum weight element of $\mathtt{M}(\sigma, e)$ only needs to be performed over the set of planar trees. This search can be executed using a dynamic program, outlined in Alg. 5. The algorithm finds the minimum weight MLT converting $\langle 1 \cdots k\rangle$ by first finding the minimum weight of MLTs of shorter cycles of the form $\langle i \cdots j\rangle$, where $1 \le i < j \le k$. Lemma 3.23 establishes that Alg. 5 produces a minimum weight MLT in $\mathtt{M}(\sigma, e)$.

Lemma 3.23. *The output weight of Alg. 5, $C(1,k)$, equals $\mathsf{L}_\varphi(\sigma, e)$.*

Proof. We look at the computations performed in the algorithm from a top-down point of view.

Let $C_T(i,j)$ be the weight of the transform converting $\sigma^{i,j}$ to e, where $\sigma^{i,j} = \langle i \cdots j\rangle$, using the edges of $T[\{i,\ldots,j\}]$, where T is an arbitrary planar spanning tree over the vertices $\{1,\ldots,k\}$ arranged on a circle. For a fixed T, let r and s be defined as in the proof of Lemma 3.21. We may write

$$\langle i \cdots j\rangle = \langle s+1 \cdots r\rangle \langle i\ r\rangle \langle r \cdots j\rangle \langle i \cdots s\rangle \tag{3.30}$$

where $i \le s < r \le j$. Thus

$$C_T(i,j) = C_T(s+1,r) + \varphi^*_{\langle i\ r\rangle} + C_T(r,j) + C_T(i,s). \tag{3.31}$$

Define $C(i,j) = C_{T^*}(i,j)$, where

$$T^* = \arg\min_T C_T(i,j)$$

denotes a tree that minimizes the weight of the transform converting $\langle i \cdots j\rangle$ to e. Now

$$C(i,j) = C(s^*+1, r^*) + \varphi^*_{\langle i\ r^*\rangle} + C(r^*, j) + C(i, s^*), \tag{3.32}$$

where s^* and r^* are the values that minimize the right side of (3.31) under the

constraint $1 \le i \le s < r \le j$. Hence, $C(i,j)$ can be obtained recursively, with the initialization

$$C(i, i+1) = \varphi^*_{\langle i\ i+1\rangle}. \tag{3.33}$$

The algorithm searches over s and r and computes $C(1,k)$ using (3.32) and (3.33).

Although these formulas are written in a recursive form, Alg. 5 is described as a dynamic program. The algorithm first computes $C(i,j)$ for small values of i and j; then it finds the weight for longer cycles. That is, for each $2 \le l \le k-1$, in increasing order, $C(i, i+l)$ is computed by choosing its optimal transform in terms of weights of transforming smaller cycles. □

Example 3.24. As an example, let us find $\mathsf{L}_\varphi(\sigma, e)$ for $\sigma = \langle 1\ 2\ 3\ 4\rangle$ using the above algorithm. Let Φ be the matrix of transposition weights, with $\Phi_{ij} = \varphi_{\langle i\ j\rangle}$. After optimizing the transposition weights in Φ via Alg. 2, we obtain Φ^*, shown next to Φ.

$$\Phi = \begin{bmatrix} 0 & 5 & 10 & 3 \\ - & 0 & 2 & 3 \\ - & - & 0 & 9 \\ - & - & - & 0 \end{bmatrix}, \qquad \Phi^* = \begin{bmatrix} 0 & 5 & 9 & 3 \\ - & 0 & 2 & 3 \\ - & - & 0 & 7 \\ - & - & - & 0 \end{bmatrix}. \tag{3.34}$$

From Alg. 5, we obtain

$$\begin{aligned} C(1,3) &= C(2,3) + \varphi^*_{\langle 1\ 2\rangle} = 7, && (s,r) = (1,2), \\ C(2,4) &= C(2,3) + \varphi^*_{\langle 2\ 4\rangle} = 5, && (s,r) = (3,4). \end{aligned}$$

Now consider the cycle $\langle 1\ 2\ 3\ 4\rangle$, for which $i = 1$ and $j = 4$. The algorithm compares $\binom{4}{2} = 6$ ways to represent the weight of this cycle using the weight of shorter cycles. The minimum weight is obtained by choosing $s = 2$ and $r = 4$, so that

$$C(1,4) = C(2,4) + \varphi^*_{\langle 1\ 4\rangle} = 8.$$

Writing C as a matrix, where $C(i,j) = C_{ij}$, we have:

$$C = \begin{bmatrix} 0 & 5 & 7 & 8 \\ - & 0 & 2 & 5 \\ - & - & 0 & 7 \\ - & - & - & 0 \end{bmatrix}.$$

We can modify the above algorithm to also find the underlying MLT by using (3.30) to write the transform of every cycle with respect to r and s that minimize the weight of the transform. For example, from (3.30), by substituting the appropriate values of r and s, we obtain

$$\langle 1\ 2\ 3\ 4\rangle = \langle 2\ 3\ 4\rangle \langle 1\ 4\rangle = \langle 3\ 4\rangle \langle 2\ 4\rangle \langle 1\ 4\rangle.$$

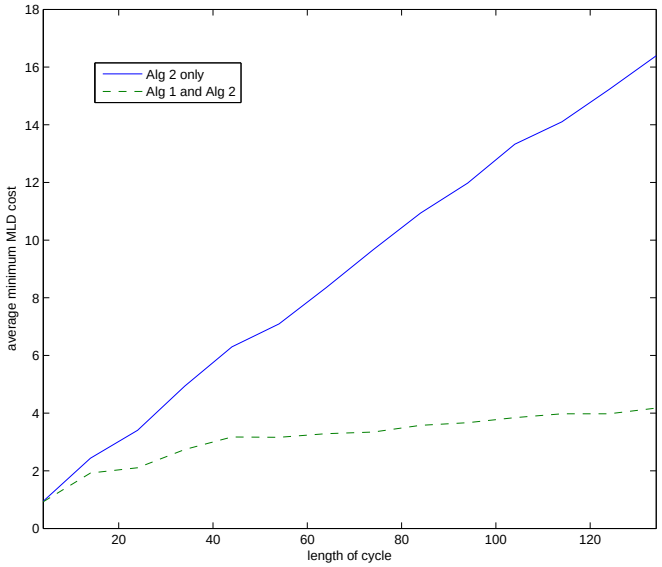


Figure 3.15: The average minimum MLT weight vs the length of the cycle. Transposition weights are chosen independently and uniformly in $[0, 1]$.

Thus,

$$\arg \min_{\tau \in \mathbb{M}(\langle 1\ 2\ 3\ 4 \rangle, e)} \mathrm{wt}(\tau) = (\langle 1\ 4 \rangle, \langle 2\ 4 \rangle, \langle 3\ 4 \rangle).$$

□

Computational Complexity: The initialization steps are performed in $O(k)$ time. The algorithm performs a constant number of steps for each i, j, r, and s such that $1 \leq i \leq s < r \leq j \leq k$. Hence, the computational cost of the algorithm is $O\left(k^4\right)$.

Note that Alg. 5 operates on the optimized weight function φ^*, obtained as the output of Alg. 2. Figure 3.15 illustrates the importance of first reducing individual transposition weights using Alg. 2 before applying the dynamic program. Since the dynamic program can only use $k-1$ transpositions of minimum weight, it cannot optimize the individual weights of transpositions and strongly relies on the reduction of Alg. 2 for producing low weight solutions. In Figure 3.15, the transposition weights were chosen independently according to a uniform distribution over $[0, 1]$.

3.5.2 Relationship between d_φ and L_φ

For the cycle $\sigma = \langle 1 \cdots k \rangle$, consider the MLT

$$\tau = (\langle k\ k-1 \rangle, \langle k-1\ k-2 \rangle, \ldots, \langle 2\ 1 \rangle). \tag{3.35}$$

with weight $\sum_{i=1}^{k-1} \varphi^*_{\langle i\ i+1\rangle}$. Since $\varphi^*_{\langle a\ b\rangle} = \mathsf{d}_\varphi(\langle a\ b\rangle, e)$, Lemma 3.4 implies that

$$\mathrm{wt}(\tau) \le 2\sum_{i=1}^{k-1} \mathrm{wt}(p^*_\varphi(i, i+1)) \le 2\sum_{i\in[k]} \mathrm{wt}(p^*_\varphi(i, \sigma(i))).$$

By definition of L_φ, we have $\mathsf{L}_\varphi(\sigma, e) \le \mathrm{wt}(\tau)$. Thus,

$$\mathsf{L}_\varphi(\sigma, e) \le 2\sum_{i\in[k]} \mathrm{wt}(p^*_\varphi(i, \sigma(i))).$$

Consider a permutation $\pi \in \mathbb{S}_n$. The inequality (3.5.2) holds for every cycle of π. Since the minimum weight MLT of π can be obtained by concatenating the minimum weight MLTs of its cycles, we have

$$\mathsf{L}_\varphi(\pi, e) \le 2\sum_{i\in[n]} \mathrm{wt}(p^*_\varphi(i, \pi(i))).$$

Furthermore, Lemma 3.13, implies that

$$\mathsf{d}_\varphi(\pi, e) \ge \frac{1}{2}\sum_{i\in[n]} \mathrm{wt}(p^*_\varphi(i, \pi^{-1}(i))) = \frac{1}{2}\sum_{i\in[n]} \mathrm{wt}(p^*_\varphi(i, \pi(i))).$$

Hence, we have the following theorem.

Theorem 3.25. *For a permutation $\pi \in \mathbb{S}_n$ and a weight function φ, $\mathsf{L}_\varphi(\pi, e) \le 4\mathsf{d}_\varphi(\pi, e)$.*

While Theorem 3.25 states that L_φ is a 4-approximation for d_φ, in practice, it may provide a significantly better result as demonstrated by the following example.

Example 3.26. Consider the cycle $\sigma = \langle 1\ 2\ 3\ 4\ 5\rangle$ and the weight function φ, with $\varphi_{\langle 2\ 4\rangle} = \varphi_{\langle 2\ 5\rangle} = \varphi_{\langle 3\ 5\rangle} = 1$, and $\varphi_{\langle i\ j\rangle} = 100$ for all remaining transposition $\langle i\ j\rangle$. From Lemma 3.13,

$$\mathsf{d}_\varphi(\sigma, e) \ge \frac{1}{2}(100 + 2 + 3 + 2 + 100) = 103.5.$$

For example, the second term in the sum corresponds to a path going from 2 to 5 and then from 5 to 3. The weight of this path is 2.

Since $\mathsf{d}_\varphi(\sigma, e)$ has to be an integer, it follows that $\mathsf{d}_\varphi(\sigma, e) \ge 104$.

The optimized weight function φ^*, obtained from Alg. 2, equals

$$\varphi^*_{\langle i\ j\rangle} = \begin{cases} 1, & \langle i\ j\rangle \in \{\langle 2\ 5\rangle, \langle 3\ 5\rangle, \langle 2\ 4\rangle\} \\ 3, & \langle i\ j\rangle \in \{\langle 2\ 3\rangle, \langle 4\ 5\rangle\} \\ 5, & \langle i\ j\rangle = \langle 3\ 4\rangle \\ 100, & \text{otherwise} \end{cases}$$

A minimum weight MLT can be computed using the dynamic program of

Alg. 5. One minimum weight MLT equals $\tau_L = (\langle 2\ 5\rangle, \langle 1\ 2\rangle, \langle 3\ 5\rangle, \langle 4\ 5\rangle)$, and has weight $\mathsf{L}_\varphi(\sigma, e) = 105$.

Hence, the inequality $\mathsf{L}_\varphi(\sigma, e) \leq 4\mathsf{d}_\varphi(\sigma, e)$ holds. Furthermore, note that σ is an even cycle, and hence must have an even number of transpositions in any of its transforms. This shows that $\mathsf{L}_\varphi(\sigma, e) = \mathsf{d}_\varphi(\sigma, e) = 105$. □

3.5.3 Merging cycles

In the previous subsection, we demonstrated that the minimum weight of an MLT for a cycle represents a constant approximation for d_φ. The MLT of a permutation is the concatenation of the MLTs of its individual cycles. Instead of concatenating the minimum weight MLTs of the cycles of a permutation, one may "merge" all the cycles and find the minimum weight MLT for the resulting cycle. Such an approach may lead to a better approximation of d_φ. For example, it may happen that the weight of transpositions within each cycle are much higher than the weights of transpositions between elements in different cycles. It is therefore useful to analyze how merging of cycles may affect the overall weight of a minimum weight MLT.

We propose a simple merging method that consists of two steps:

1. For a permutation π with k cycles, find a sequence of transpositions

$$\tau' = (\tau'_1, \ldots, \tau'_{k-1})$$

   so that $\sigma' = \pi\tau'_1 \cdots \tau'_{k-1}$ is a single cycle. Ideally, one should choose τ' such that it has minimum possible weight, although this is not required in the proofs to follow.

2. Find the minimum weight MLT τ of σ'. The transform

$$\tau'' = (\tau'_1, \ldots, \tau'_{k-1}, \tau_1, \ldots, \tau_{n-1})$$

   is a transform converting π to e and its weight is an upper-bound on $\mathsf{d}_\varphi(\pi, e)$.

Each τ'_i is a transposition merging two cycles. The weight of τ' equals $\sum_{i=1}^{k-1} \varphi^*_{\langle a_i\ b_i\rangle}$, where $\tau'_i = \langle a_i\ b_i\rangle$. The weight of the transform τ'' equals

$$\mathrm{wt}(\tau'') = \sum_{i=1}^{k-1} \varphi^*_{\langle a_i\ b_i\rangle} + \mathsf{L}_\varphi(\sigma', e). \tag{3.36}$$

Furthermore, we have

$$\mathsf{L}_\varphi(\sigma', e) \leq 4\mathsf{d}_\varphi(\sigma', e) \leq 4\left(\sum_{i=1}^{k-1} \varphi^*_{\langle a_i\ b_i\rangle} + \mathsf{d}_\varphi(\pi, e)\right), \tag{3.37}$$

where the second inequality follows from the fact that $\sigma' = \pi\tau'_1 \cdots \tau'_{k-1}$. Hence, from (3.36) and (3.37), we find

$$\begin{aligned} \mathrm{wt}(\tau'') &\le \sum_{i=1}^{k-1} \varphi^*_{\langle a_i \ b_i \rangle} + 4\left(\sum_{i=1}^{k-1} \varphi^*_{\langle a_i \ b_i \rangle} + \mathsf{d}_\varphi(\pi, e) \right) \\ &\le 5k\varphi^*_{max} + 4\mathsf{d}_\varphi(\pi, e), \end{aligned} \tag{3.38}$$

where φ^*_{max} is the highest weight in φ^*. The approximation ratio α, defined as $\mathrm{wt}(\tau'')/\mathsf{d}_\varphi(\pi, e)$, is bounded as

$$\alpha \le 4 + \frac{5k}{n-k} \frac{\varphi^*_{max}}{\varphi^*_{min}} = 4 + \frac{5k/n}{1-k/n} \frac{\varphi^*_{max}}{\varphi^*_{min}}, \tag{3.39}$$

which follows from the fact $\mathsf{d}_\varphi(\pi, e) \ge (n-k)\, \varphi^*_{min}$, where φ^*_{min} is the smallest weight in φ^*, assumed to be nonzero.

Although α, according to the expression above, is bounded by a value strictly larger than four, this does not necessarily imply that merging cycles is suboptimal compared to running the minimum weight MLT algorithm on individual cycles. Furthermore, if the minimum weight transform for single cycles can be computed correctly, instead of τ one can use a transform with weight $\mathsf{d}_\varphi(\sigma', e)$. Then,

$$\begin{aligned} \mathrm{wt}(\tau'') &= \sum_{i=1}^{k-1} \varphi^*_{\langle a_i \ b_i \rangle} + \mathsf{d}_\varphi(\sigma', e). \\ &\le 2 \sum_{i=1}^{k-1} \varphi^*_{\langle a_i \ b_i \rangle} + \mathsf{d}_\varphi(\pi, e) \\ &\le 2k\varphi^*_{max} + \mathsf{d}_\varphi(\pi, e). \end{aligned}$$

The approximation ratio in this case is bounded as

$$\alpha \le 1 + \frac{2k}{n-k} \frac{\varphi^*_{max}}{\varphi^*_{min}} = 1 + \frac{2k/n}{1-k/n} \frac{\varphi^*_{max}}{\varphi^*_{min}}.$$

Lemma 3.27. *Let π be a randomly chosen permutation from $\mathbb{S}_n$. Given that the minimum weight transform for single cycles can be computed correctly, and provided that $\varphi^*_{max}/\varphi^*_{min} = o(n/\log n)$, α converges to one in probability as $n \to \infty$.*

Proof. Let X_n be the random variable denoting the number of cycles in a random permutation $\pi_n \in \mathbb{S}_n$. It is well known that $EX_n = \sum_{j=1}^{n} \frac{1}{j} = H(n)$ and that $EX_n(X_n - 1) = (EX_n)^2 - \sum_{j=1}^{n} \frac{1}{j^2}$ [63]. Here, $H(n)$ denotes the nth Harmonic number. Thus

$$EX_n^2 = O\left((\ln n)^2 \right), \tag{3.40}$$

which shows that $(X_n/n)(\varphi^*_{max}/\varphi^*_{min}) \to 0$ in quadratic mean as $n \to \infty$. Hence

$(X_n/n)(\varphi^*_{max}/\varphi^*_{min}) \to 0$ in probability. By Slutsky's theorem [64], $\alpha \to 1$ in probability as $n \to \infty$. □

In the following example, all operations are performed modulo 10, with zero replaced by 10.

Example 3.28. Consider the permutation $\pi = \sigma_1\sigma_2$, where $\sigma_1 = \langle 1\ 7\ 3\ 9\ 5\rangle$ and $\sigma_2 = \langle 2\ 8\ 4\ 10\ 6\rangle$, and the weight function φ,

$$\varphi(i,j) = \begin{cases} 1, & d(i,j) = 1 \\ \infty, & \text{otherwise} \end{cases}$$

where $d(i,j) = \min\{|i-j|, 10-|i-j|\}$.

Note that $\mathrm{wt}\left(p^*_\varphi(i,j)\right) = d(i,j)$, We make the following observations regarding the transforms of π.

1. Minimum weight transform: Since we currently do not know of an efficient enough algorithm for finding the minimum weight transform of a permutation, we were not able to find the minimum weight transform of π. Nevertheless, using Theorem 3.14, one can obtain the following bounds:

$$\mathsf{d}_\varphi(\pi, e) \geq \frac{1}{2}\sum_{i=1}^{10} \mathrm{wt}\left(p^*\left(i, \pi(i)\right)\right) = \frac{2\cdot 5\cdot 4}{2} = 20, \tag{3.41}$$

$$\mathsf{d}_\varphi(\pi, e) \leq 4 \cdot 20 = 80. \tag{3.42}$$

2. Minimum weight MLT: We have $\varphi^*_{\langle i\ j\rangle} = 2d(i,j) - 1$. The minimum weight MLTs for the cycles σ_1 and σ_2 are

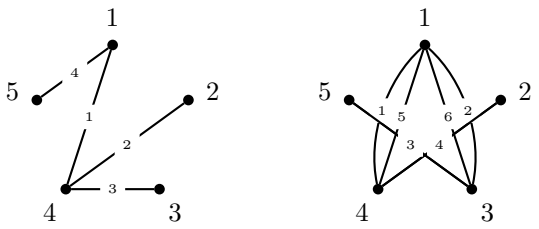
$$\begin{aligned} \tau^{(1)} &= \left(\langle 9\ 5\rangle, \langle 1\ 3\rangle, \langle 3\ 7\rangle, \langle 1\ 9\rangle\right), \\ \tau^{(2)} &= \left(\langle 6\ 10\rangle, \langle 2\ 4\rangle, \langle 4\ 8\rangle, \langle 2\ 10\rangle\right), \end{aligned}$$

   respectively, each of weight 20. A minimum weight MLT of π is the concatenation of the minimum weight MLTs of σ_1 and $\sigma_2$with overall weight equal to 40.

3. Merging cycles: We let $\tau' = \langle 5\ 6\rangle$ to merge the cycles and obtain $\sigma' = \pi\langle 5\ 6\rangle = \langle 1\ 7\ 3\ 9\ 5\ 2\ 8\ 4\ 10\ 6\rangle$. The minimum weight of an MLT of σ' can be shown to be 37. Since the weight of the transposition $\langle 5\ 6\rangle$ must also be accounted for, the total weight is 38. Observe that this weight is smaller than that of part 2, and hence merging cycles may provide better solutions than indicated by (3.39) or obtained by concatenating minimum weight MLTs.

Putting the parts together, we have

$$20 \leq \mathsf{d}_\varphi(\pi, e) \leq 38.$$



(a) Min weight MLT

(b) Min weight transform

Figure 3.16: The minimum weight MLT (a), and the minimum weight transform (b), for $\pi = \langle 1\ 2\ 3\ 4\ 5 \rangle$. Edge labels denote the order in which transpositions are applied.

□

3.5.4 Metric-path and Extended-path Weight Functions

By inspecting the proof of Lemma 3.15, one can see that the transform described in Remark 3.16, for a permutation π with respect to a metric-path weight function φ, is an MLT. Thus its weight, which equals $\mathsf{d}_\varphi(\pi, e)$, is an upper-bound on $\mathsf{L}_\varphi(\pi, e)$. It follows, for every permutation π and a metric-path weight function φ, that $\mathsf{L}_\varphi(\pi, e) = \mathsf{d}_\varphi(\pi, e)$ and that every minimum weight MLT of π is also a minimum weight transform converting π to e.

A similar argument, using Lemma 3.17 and Remark 3.18, implies that $\mathsf{L}_\varphi(\pi, e) \leq 2\mathsf{d}_\varphi(\pi, e)$ for every permutation π and an extended-path weight function φ.

Example 3.29. Consider the cycle $\pi = \langle 1\ 2\ 3\ 4\ 5 \rangle$ and the extended-path weight function with $\Theta_s = (2, 4, 1, 3, 5)$ where the weight of each edge of Θ_s is 1.

By inspection, one can see that $(\langle 1\ 4 \rangle, \langle 1\ 3 \rangle, \langle 3\ 5 \rangle, \langle 2\ 4 \rangle, \langle 1\ 4 \rangle, \langle 1\ 3 \rangle)$ is a minimum weight transform converting π to e with weight $\mathsf{d}_\varphi(\pi, e) = 6$. A minimum weight MLT of π is $(\langle 1\ 4 \rangle, \langle 2\ 4 \rangle, \langle 3\ 4 \rangle, \langle 1\ 5 \rangle)$, with weight $\mathsf{L}_\varphi(\pi, e) = 8$. We observe that the inequality $\mathsf{L}_\varphi(\pi, e) \leq 2\mathsf{d}_\varphi(\pi, e)$ is satisfied. The transforms are shown in Figure 3.16 □

3.6 Examples of Aggregation with Similarity Distance

In this section, we present examples of rank aggregation using the similarity distance, which is equivalent to the weighted transposition distance applied to inverse rankings. In particular, we aim to show that the weighted transposition distance may provide more diverse aggregates when the votes themselves are diverse, as compared to the Kendall τ distance.

Finding the aggregate, discussed in more detail in the next chapter, is a computationally difficult problem. Since our goal is to demonstrate the effect of

the similarity distance on aggregation, computations in the following examples are performed via exhaustive search techniques. Thus all solutions are exact.

Example 3.30. Consider the votes listed in Σ below:

$$\Sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ \hline 3 & 2 & 1 & 4 \\ \hline 4 & 1 & 3 & 2 \end{pmatrix}.$$

Suppose that even numbers and odd numbers represent different types of candidates in a way that the following weight function is appropriate:

$$\varphi_{\langle i\ j\rangle} = \begin{cases} 1, & \text{if } i,j \text{ are both odd or both even,} \\ 2, & \text{else.} \end{cases}$$

Note that the votes are "diverse" in the sense that they alternate between odd and even numbers. On the other hand, the Kemeny aggregate is $(1,3,2,4)$, which puts all odd numbers ahead of all even numbers. Aggregation using the similarity distance described above yields $(1,2,3,4)$, a solution which may be considered "diverse" since the even and odd numbers alternate in the solution. The reason behind this result is that the Kemeny optimal solution is oblivious to the identity of the candidates and their (dis)similarities, while aggregation based on similarity distances take such information into account.

This example, as well as the following one, can also be viewed in the context of assignment aggregation. Suppose there are 4 jobs and 4 candidates and a committee with three members in tasked with assigning candidates to jobs. Each committee member presents its assignment in the form of a permutation; job 1 is assigned to the first candidate in the permutation, job 2 to the second candidate, and so on. For the votes Σ and the weight function φ, given above, the assignment aggregate using the similarity distance is $(1,2,3,4)$. The weight function φ is appropriate here if, for example, candidates 1 and 3 have the same expertise and so do candidates 2 and 4.

Example 3.31. Consider the votes listed in Σ below:

$$\Sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 3 & 6 & 5 & 2 & 1 & 4 \\ \hline 3 & 6 & 5 & 2 & 1 & 4 \\ \hline 5 & 4 & 1 & 6 & 3 & 2. \end{pmatrix}.$$

Suppose that the weight function is the same as the one used in the previous example. In this case, neither the Kemeny aggregates nor the weighted transposition distance aggregates are unique. More precisely, Kendall τ gives four

solutions:

$$\left(\begin{array}{cccccc} 3 & 5 & 1 & 6 & 2 & 4 \\ \hline 3 & 5 & 1 & 2 & 4 & 6 \\ \hline 1 & 3 & 5 & 2 & 4 & 6. \\ \hline 1 & 3 & 5 & 6 & 2 & 4. \end{array}\right),$$

while there exist nine optimal aggregates under the weighted transposition distance of the previous example, of total distance 10:

$$\left(\begin{array}{cccccc} 5 & 6 & 3 & 4 & 1 & 2 \\ \hline 5 & 4 & 3 & 2 & 1 & 6 \\ \hline 5 & 2 & 3 & 6 & 1 & 4 \\ \hline 3 & 4 & 1 & 2 & 5 & 6 \\ \hline 3 & 6 & 1 & 4 & 5 & 2 \\ \hline 3 & 2 & 1 & 6 & 5 & 4 \\ \hline 1 & 4 & 5 & 2 & 3 & 6 \\ \hline 1 & 2 & 5 & 6 & 3 & 4 \\ \hline 1 & 6 & 5 & 4 & 3 & 2. \end{array}\right).$$

Note that *none of* the Kemeny optimal aggregates have good diversity properties: the top half of the rankings consists exclusively of odd numbers. On the other hand, the optimal weighted transposition rankings *all contain* exactly one even element among the top-three candidates. Such diversity properties are hard to prove theoretically.

Chapter 4

Aggregation Algorithms

4.1 Introduction

In this chapter, we study rank aggregation algorithms that can be used on weighted distances. Recall that rank aggregation refers to finding a ranking that is a "representative" of a set of votes. The representative ranking is termed the *aggregate ranking*. Throughout this chapter, we use m to denote the number of votes and use $\Sigma = \{\sigma_1, \ldots, \sigma_m\}$ to denote the multiset of the votes. Our focus is on distance-based rank aggregation where the aggregate is given by

$$\arg\min_{\pi} \sum_{\sigma \in \Sigma} \mathsf{d}(\pi, \sigma), \tag{4.1}$$

for a given distance function d.

It is known that computing the solution to (4.1) for the Kendall τ distance, i.e., computing the Kemeny aggregate, is NP-hard [46]. Since the Kendall τ distance is a special case of the weighted Kendall distance and the weighted transposition distance, finding the aggregate ranking for those distances is also NP-hard. In particular, exhaustive search approaches – akin to the one we used in Examples 2.14-2.17 – are not computationally feasible for large problems.

However, assuming that π^* is the solution to (4.1), the ranking $\sigma \in \Sigma$ closest to π^* provides a 2-approximation for the aggregate ranking. This easily follows from the fact that the Kendall τ distance satisfies the triangle inequality. As a result, one only has to evaluate the pairwise distances of the votes in Σ, which can be done in polynomial time, in order to identify a 2-approximation aggregate for the problem. Assuming that the weighted distance under consideration can be computed efficiently (for example, in the case of the weighted Kendall distance with a monotonic weight function), the aggregate ranking for the weighted distance can also be computed in polynomial time as the weighted distance is also a metric and thus satisfies the triangle inequality.

In this chapter, we present two algorithms for finding the aggregate when the relative significance of different positions in rankings is indicated via a weight function. The first algorithm is the combination of an approximation algorithm

This chapter contains material from the work [65], coauthored with B. Touri and O. Milenkovic.

based on bipartite matching and a local search method. The second algorithm relies on the stationary probabilities of a Markov chain obtained from the votes, similar to the PageRank algorithm.

4.2 Rank Aggregation Using Bipartite Matching

For any distance function that may be written as

$$\mathsf{d}(\pi,\sigma) = \sum_{k=1}^{n} f(\pi^{-1}(k),\sigma^{-1}(k)), \tag{4.2}$$

where f denotes an arbitrary nonnegative function, one can find an *exact solution* to (4.1) using a minimum weight bipartite matching algorithm. Consider a weighted complete bipartite graph with vertex sets X and Y, where $X = \{1,\dots,n\}$ corresponds to the n ranks to be filled in, and $Y = \{1,\dots,n\}$ corresponds to the n candidates. We say that a perfect bipartite matching M corresponds to a permutation π whenever $(ij) \in M$ if and only if $\pi(i) = j$ for all $i \in X$ and $j \in Y$. If we assign candidates j to rank i, then the contribution to the objective function of (4.1) is $\sum_{l=1}^{m} f(i,\sigma_l^{-1}(j))$. Hence, if the weight assigned to each edge (ij), for $i \in X$ and $j \in Y$, equals

$$\sum_{l=1}^{m} f(i,\sigma_l^{-1}(j)),$$

then the minimum weight perfect matching corresponds to a solution of (4.1).

The following distance functions are of the form given in (4.2):

- Spearman's footrule distance with $f(x,y) = |x-y|$.
- Spearman's rank correlation with $f(x,y) = (x-y)^2$.
- The Hamming distance with $f(x,y) = 0$ if $x = y$, and $f(x,y) = 1$ if $x \neq y$.
- The D_φ distance with $f(x,y) = \mathrm{wt}(p^*_\varphi(x,y))$.
- The Weighted transposition distance for a metric-path weight function φ, since it is a special case of D_φ.

The matching algorithm approach was first proposed in [46] as a 2-approximation for classical Kendall τ aggregation. We extend this approach to aggregation based on the weighted Kendall distance. The idea is to approximate the weighted Kendall distance for a weight function $\varphi : \mathbb{A}_n \to \mathbb{R}_{\geq 0}$ using D_φ as demonstrated by the following proposition.

Proposition 4.1. *Let*

$$\pi^* = \arg\min_{\pi} \sum_{l=1}^{m} \mathsf{d}_\varphi(\pi,\sigma_i)$$

be the aggregate with respect to the weighted Kendall distance d_φ *for a weight function* $\varphi : \mathbb{A}_n \to \mathbb{R}_{\geq 0}$*, and suppose*

$$\pi' = \arg\min_{\pi} \sum_{l=1}^{m} D_\varphi(\pi, \sigma_i).$$

The permutation π' *is a 2-approximation for* π^* *in the sense that*

$$\sum_{l=1}^{m} \mathsf{d}_\varphi(\pi', \sigma_i) \leq 2 \sum_{l=1}^{m} \mathsf{d}_\varphi(\pi^*, \sigma_i).$$

Proof. From Lemma 2.13, for a weight function $\varphi : \mathbb{A}_n \to \mathbb{R}_{\geq 0}$ and for permutations π and σ,

$$\frac{1}{2} D_\varphi(\pi, \sigma) \leq \mathsf{d}_\varphi(\pi, \sigma) \leq D_\varphi(\pi, \sigma).$$

Thus we have

$$\sum_{l=1}^{m} \mathsf{d}_\varphi(\pi', \sigma_i) \leq \sum_{l=1}^{m} D_\varphi(\pi', \sigma_i).$$

and

$$\frac{1}{2} \sum_{l=1}^{m} D_\varphi(\pi^*, \sigma_i) \leq \sum_{l=1}^{m} \mathsf{d}_\varphi(\pi^*, \sigma_i).$$

From the optimality of π' with respect to D_φ, we find

$$\sum_{l=1}^{m} D_\varphi(\pi', \sigma_i) \leq \sum_{l=1}^{m} D_\varphi(\pi^*, \sigma_i).$$

Hence

$$\sum_{l=1}^{m} \mathsf{d}_\varphi(\pi', \sigma_i) \leq 2 \sum_{l=1}^{m} \mathsf{d}_\varphi(\pi^*, \sigma_i).$$

□

In fact, the above proposition applies to the larger class of weighted transposition distances with extended-path weights. It can similarly be shown that for a weighted transposition distance with general weights, π' is a 4-approximation. Finally, for a weighted transposition distance with a metric weight function, π' represents a 2-approximation.

A simple approach for improving the performance of the matching based algorithm is to couple it with a local descent method. Assume that an estimate of the aggregate at step ℓ equals $\pi^{(\ell)}$. Then

$$\pi^{(\ell+1)} = \pi^{(\ell)} \arg\min_{\tau \in \mathbb{A}_n} \sum_{i=1}^{m} \mathsf{d}(\pi^{(\ell)} \tau, \sigma_i).$$

The search terminates when the cumulative distance of the aggregate from the set of votes Σ cannot be decreased further. We choose the starting point $\pi^{(0)}$ to be the ranking π' of Prop. 4.1 obtained by the minimum weight bipartite matching algorithm. This method will henceforth be referred to as Bipartite Matching with Local Search (BMLS).

An important question at this point is how does the approximate nature of the BMLS aggregation process changes the aggregate, especially with respect to the top-vs-bottom or similarity property? This question is hard, and we currently have no mathematical results pertaining to this problem. Instead, we describe a number of simulation results that may guide the future analysis of this issue.

In order to see the effect of the BMLS on vote aggregation, we revisit Examples 2.14-2.17. In *all except for one case* the solution provided by BMLS is the same as the exact solution, both for the Kendall τ and weighted Kendall distances.

The exception is Example 2.15. In this case, for the weight function $\varphi_{\langle i\ i+1\rangle} = (2/3)^{i-1}, i \in [3]$, the exact solution equals $(1,4,2,3)$ but the solution obtained via BMLS equals $(4,2,3,1)$. Note that these two solutions differ significantly in terms of their placement of candidate 1, ranked first in the exact ranking and last in the approximate ranking. The distances between the two solutions, $\mathsf{d}_\varphi((1,4,2,3),(4,2,3,1))$, equals 2.11 and is rather large. Nevertheless, the cumulative distances to the votes are very close in value:

$$\sum_i \mathsf{d}_\varphi((1,4,2,3),\sigma_i) = 9,$$
$$\sum_i \mathsf{d}_\varphi((4,2,3,1),\sigma_i) = 9.11.$$

Hence, as with any other distance based approach, the approximation result may sometimes diverge significantly from the optimum solution while the closeness of the approximate solution to the set of votes is nearly the same as that of the optimum solution.

4.3 Vote Aggregation Using PageRank

An algorithm for data fusion, based on the PageRank [66] and HITS [67] algorithms for ranking web pages, was proposed in [46]. PageRank is one of the most important algorithms developed for search engines used by Google, with the aim of scoring webpages based on their relevance. Each webpage is both a voter and a candidate. A link from one webpage to another indicates an approval vote of the former for the latter. The rank of each webpage not only depends on the number of the webpages that contain a link to it but also on their rank: A vote from a highly ranked webpage is more influential than a vote from a lowly ranked page. A simplified version of PageRank can be modeled as a "random surfer." The random surfer performs a random walk on the graph of webpages where links are modeled as directed edges. Note that this process represents a Markov chain in which transition probabilities are indicated by the hyperlinks. Ranking of the webpages is obtained by computing the stationary probabilities of the position of the random surfer. Thus webpages with many

links to them are ranked high as well as webpages that have links from webpages that are ranked high.

This idea can be easily adapted to the rank aggregation problem in several different settings. In such an adaptation, the states of a Markov chain correspond to the candidates and the transition probabilities are functions of the votes. Dwork et al. [12, 46] proposed four different ways for computing the transition probabilities from the votes. Below, we describe the method that is most suitable for our problem and provide a generalization of the algorithm for weighted distance aggregation.

Consider a Markov chain with states indexed by the candidates. Let P denote the transition probability matrix of the Markov chain, with P_{ij} denoting the probability of going from state (candidate) i to state j. In [46], the transition probabilities are evaluated as

$$P_{ij} = \frac{1}{m} \sum_{\sigma \in \Sigma} P_{ij}(\sigma),$$

where

$$P_{ij}(\sigma) = \begin{cases} \frac{1}{n}, & \text{if } \sigma^{-1}(j) < \sigma^{-1}(i), \\ 1 - \frac{\sigma^{-1}(i)-1}{n}, & \text{if } i = j, \\ 0, & \text{if } \sigma^{-1}(j) > \sigma^{-1}(i). \end{cases}$$

Our Markov chain model for weighted Kendall distance is similar, with a modification that includes incorporating transposition weights into the transition probabilities. To accomplish this task, we proceed as follows.

Let $w_k = \varphi_{\langle k\ k+1 \rangle}$, and let $i_\sigma = \sigma^{-1}(i)$ for candidate $i \in [n]$. We set

$$\beta_{ij}(\sigma) = \begin{cases} \max\limits_{l : j_\sigma \le l < i_\sigma} \frac{\sum_{h=l}^{i_\sigma - 1} w_h}{i_\sigma - l}, & \text{if } j_\sigma < i_\sigma, \\ \sum\limits_{k : k_\sigma > i_\sigma} \beta_{ki}(\sigma), & \text{if } j_\sigma = i_\sigma, \\ 0, & \text{if } j_\sigma > i_\sigma. \end{cases} \tag{4.3}$$

The transition probabilities equal

$$P_{ij} = \frac{1}{m} \sum_{k=1}^{m} P_{ij}(\sigma_k),$$

with

$$P_{ij}(\sigma) = \frac{\beta_{ij}(\sigma)}{\sum_k \beta_{ik}(\sigma)}.$$

Intuitively, the transition probabilities described above may be interpreted as follows. The transition probabilities are obtained by averaging the transition probabilities corresponding to individual votes $\sigma \in \Sigma$. Suppose that a vote σ is of the form $(\dots, k, j, i, \dots)$. Note that $j_\sigma = i_\sigma - 1$ and $k_\sigma = i_\sigma - 2$. The probability of going from candidate i to candidate j is proportional to $w_{j_\sigma} = \varphi_{\langle j_\sigma\ i_\sigma \rangle}$. This implies that if $w_{j_\sigma} > 0$, one moves from candidate i to candidate j with positive

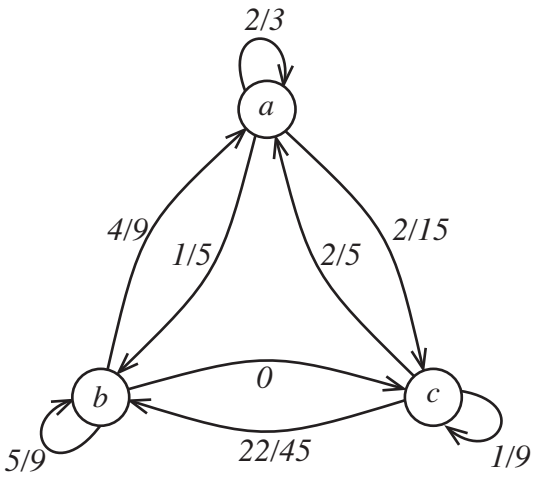


Figure 4.1: The Markov chain for Example 4.2.

probability. Furthermore, larger values for w_{j_σ} result in higher probabilities for moving from i to j.

In the case of candidate k, it seems reasonable to let the probability of transitioning from candidate i to candidate k be proportional to $\frac{w_{j_\sigma}+w_{k_\sigma}}{2}$. However, since k is ranked before j by σ, it is natural to require that the probability of moving to candidate k from candidate i be at least as high as the probability of moving to candidate j from candidate i. This reasoning leads to $\beta_{ik} = \max\{w_{j_\sigma}, \frac{w_{j_\sigma}+w_{k_\sigma}}{2}\}$ and motivates using the maximum in (4.3). The same reasoning applies to other candidates that are ranked before candidate i. Finally, the probability of staying with candidate i is proportional to the sum of the β's from candidates placed after candidate i.

Example 4.2. Let the votes in Σ consist of $\sigma_1 = (a, b, c)$, $\sigma_2 = (a, b, c)$, and $\sigma_3 = (b, c, a)$, and let $w = (w_1, w_2) = (2, 1)$.

Consider the vote $\sigma_1 = (a, b, c)$. We have $\beta_{ba}(\sigma_1) = \frac{w_1}{1} = 2$. Note that if w_1 is large, then β_{ba} is large as well.

In addition, $\beta_{cb}(\sigma_1) = \frac{w_2}{1} = 1$ and

$$\beta_{ca} = \max\left\{\frac{w_1 + w_2}{2}, \beta_{cb}\right\} = \frac{3}{2}.$$

The purpose of the max function is to ensure that $\beta_{ca} \geq \beta_{cb}$, which is a natural requirement given that a is ranked before b according to σ_1.

Finally, $\beta_{aa}(\sigma_1) = \beta_{ca}(\sigma_1) + \beta_{ba}(\sigma_1) = 2 + \frac{3}{2} = \frac{7}{2}$ and $\beta_{bb}(\sigma_1) = \beta_{cb}(\sigma_1) = 1$. Note that according to the transition probability model, one also has $\beta_{aa} \geq \beta_{bb}$. This may again be justified by the fact that σ_1 places a higher than b.

Since $\sigma_1 = \sigma_2$, we have

$$P(\sigma_1) = P(\sigma_2) = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{3}{5} & \frac{2}{5} & 0 \end{pmatrix}.$$

Similar computations yield

$$\begin{aligned} &\beta_{cb}(\sigma_3) = 2, \qquad \beta_{ac}(\sigma_3) = 1, \qquad \beta_{ab}(\sigma_3) = \frac{3}{2} \\ &\beta_{aa}(\sigma_3) = 0, \quad \beta_{bb}(\sigma_3) = 2 + \tfrac{3}{2} = \tfrac{7}{2}, \quad \beta_{cc}(\sigma_3) = 1, \end{aligned}$$

and thus

$$P(\sigma_3) = \begin{pmatrix} 0 & \frac{3}{5} & \frac{2}{5} \\ 0 & 1 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \end{pmatrix}.$$

From the $P(\sigma_1), P(\sigma_2)$, and $P(\sigma_3)$, we obtain

$$P = \frac{P(\sigma_1) + P(\sigma_2) + P(\sigma_3)}{3} = \begin{pmatrix} \frac{2}{3} & \frac{1}{5} & \frac{2}{15} \\ \frac{4}{9} & \frac{5}{9} & 0 \\ \frac{2}{5} & \frac{22}{45} & \frac{1}{9} \end{pmatrix}.$$

The Markov chain corresponding to P is given in Figure 4.1. The stationary distribution of this Markov chain is $(0.56657, 0.34844, 0.084986)$ which corresponds to the ranking (a, b, c).

Example 4.3. The performance of the Markov chain approach described above cannot be easily evaluated analytically, as is the case with any related aggregation algorithm proposed so far.

We hence test the performance of the scheme on examples for which the optimal solutions are easy to evaluate numerically. For this purpose, we consider a simple test example, with $m = 11$. The set of votes (rankings) is listed below

$$\Sigma^T = \left(\begin{array}{c|c|c|c|c|c|c|c|c|c|c} 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 5 \\ 2 & 2 & 2 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 4 & 4 & 4 & 4 & 5 & 5 & 3 & 3 \\ 4 & 4 & 4 & 5 & 5 & 5 & 5 & 3 & 3 & 4 & 4 \\ 5 & 5 & 5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right).$$

Note that due to the transpose operator, each column corresponds to a vote, e.g., $\sigma_1 = (1, 2, 3, 4, 5)$.

Let us focus on candidates 1 and 2. Using the plurality rule, one would arrive at the conclusion that candidate 1 should be the winner, given that 1 appears most often at the top of the list. Under a number of other aggregation rules, including Kemeny's rule and Borda's method [68], candidate 2 would be the winner.

Our immediate goal is to see how different weighted distance based rank aggregation algorithms would position candidates 1 and 2. The numerical results regarding this example are presented in Table 4.1. In the table, OPT refers to an optimal solution found by exhaustive search, and MC refers to the Markov chain method.

If the weight function is $w = (w_1, \ldots, w_4) = (1, 0, 0, 0)$, where $w_i = \varphi_{\langle i\ i+1\rangle}$, the aggregate clearly corresponds to the plurality winner; the winner is the candidate with most voters ranking her as the top candidate. A quick check of Table 4.1 reveals that all three methods identify the winner correctly. Note that the ranks of candidates other than candidate 1 obtained by the different methods are different; however, this does not affect the distance between the aggregate ranking and the votes.

The next weight function that we consider is the uniform weight function, $w = (1, 1, 1, 1)$. This weight function corresponds to the conventional Kendall τ distance. As shown in Table 4.1, all three methods produce $(2, 3, 4, 5, 1)$; the aggregates returned by BMLS and MC are optimum.

The weight function $w = (1, 1, 0, 0)$ corresponds to *ranking of the top 2* candidates. OPT and BMLS return 2 and 3 as the top two candidates, both preferring 2 to 3. The MC method, however, returns 2 and 1 as the top two candidates, with a preference for 2 over 1, and a suboptimal cumulative distance. It should be noted that this may be attributed to the fact the MC method is not designed to only minimize the average distance: another important factor in determining the winners via the MC method is that winning against strong candidates "makes one strong." In this example, candidate 1 beats the strongest candidate, candidate 2, three times, while candidate 3 beats candidate 2 only twice and this seems to be the reason for the MC algorithm to prefer candidate 1 to candidate 3. Nevertheless, the stationary probabilities of candidates 1 and 3 obtained by the MC method are very close to each other, as the vector of stationary probabilities is $(\underline{0.137}, 0.555, \underline{0.132}, 0.0883, 0.0877)$.

The weight function $w = (0, 1, 0, 0)$ corresponds to *identifying the top 2* candidates – it is not important which candidate is the first and which is the second. The OPT and BMLS identify $\{2, 3\}$ as the top two candidates.

The MC method returns the stationary probabilities $(0, 1, 0, 0, 0)$ which means that candidate 2 is an absorbing state in the Markov chain. This occurs because candidate 2 is ranked first or second by all voters. The existence of absorbing states is a drawback of the Markov chain methods. One solution is to remove 2 from the votes and re-apply MC. The MC method in this case results in the stationary distribution $(p(1), p(3), p(4), p(5)) = (0.273, 0.364, 0.182, 0.182)$, which gives us the ranking $(3, 1, 4, 5)$. Together with the fact that candidate 2 is the strongest candidate, we obtain the ranking $(2, 3, 1, 4, 5)$.

Table 4.1: The aggregate rankings and the average distance of the aggregate ranking from the votes for different weight functions w.

Method	Aggregate ranking and average distance			
	$w = (1,0,0,0)$	$w = (1,1,1,1)$	$w = (1,1,0,0)$	$w = (0,1,0,0)$
OPT	$(\underline{1},4,3,2,5)$, 0.7273	$(2,3,4,5,1)$, 2.3636	$\big(\underline{2,3},4,5,1\big)$, 1.455	$\big(\underline{3,2},5,4,1\big)$, 0.636
BMLS	$(\underline{1},2,3,4,5)$, 0.7273	$(2,3,4,5,1)$, 2.3636	$\big(\underline{2,3},1,5,4\big)$, 1.455	$\big(\underline{2,3},1,5,4\big)$, 0.636
MC	$(\underline{1},2,5,4,3)$, 0.7273	$(2,3,4,5,1)$, 2.3636	$\big(\underline{2,1},3,4,5\big)$, 1.546	$\big(\underline{2,3},1,4,5\big)$, 0.636

Chapter 5

Codes in Ulam Metric for Error-Correction in Flash Memories

5.1 Introduction

Permutation codes and permutation arrays are collections of suitably chosen codewords from the symmetric group, used in applications as varied as single user communication over Gaussian channels [69, 70], reduction of impulsive noise over power-lines [71, 72], and coding for storage [73]. Many instances of permutation-based codes were studied in the coding theory literature, with special emphasis on permutation codes in the Hamming metric (permutation arrays) and in the Kendall τ distance. The distances used for code construction in storage devices have mostly focused around two types of combinatorial measures, counting functions of adjacent transpositions and measures obtained via embeddings into the Hamming space [71, 73]. This is due to the fact that such distance measures capture the displacement of symbols in retrieved messages that arise in modern nonvolatile storage systems.

One of the most prominent emerging applications of permutation codes in storage is *rank modulation*. Rank modulation is an encoding scheme for flash memories that may improve the lifespan, storage efficiency and reliability of future generations of these storage devices [73, 76–78]. The idea behind the modulation scheme is that information should be stored in the form of rankings of the cell charges, rather than in terms of the absolute values of the charges. This simple conceptual coding framework may eliminate the problem of cell block erasures as well as potential cell over-injection issues [76, 79]. In their original formulation, rank-modulation codes represent a family of codes capable of handling errors of the form of *adjacent transpositions*. Such transposition errors represent the most likely errors in a system where the cells are expected to have nearly-uniform leakage rates. But leakage rates depend on the charge of the cells, the position of the cells and on a number of external factors, the influence of which may not be adequately captured by adjacent transposition errors. For example, if a cell for a variety of reasons has a higher leakage rate than other cells, given sufficient time, the charge of this cell may drop below the charge of a large number of other cells. Furthermore, if the number of

This chapter contains material from publications [74] and [75], coauthored with V. Skachek and O. Milenkovic.

possible charge levels is large,[1] and thus the difference between charge levels is small, a moderate charge drop may result in a significant drop in the cell's rank. One may argue that these processes may be modeled as a sequence of adjacent transposition errors. However, as this type of error is the result of a single error event, for the purpose of error correction it should be modeled as a *single error*. This is reminiscent of the scenario where one models a sequence of individual symbol errors as a single burst error [81].

In what follows, we present a novel approach to rank modulation coding which allows for correcting a more varied class of errors when compared to classical schemes. The focal point of the study is the notion of a translocation error, a concept that generalizes the notion of an adjacent transposition in a permutation. Roughly speaking, a translocation[2] moves the ranking of one particular element in the permutation below the ranking of a certain number of closest-ranked elements. As such, translocations are suitable for modeling errors that arise in flash memory systems, where high leakage levels for subsets of cells are expected or possible. Examples of such error events include errors due to radiation and breakdown of tunneling oxide, the latter being a prominent event in conventional poly-Si floating gate memories [83, 84].

A translocation may be viewed as an extension of an adjacent transposition. In addition, translocations correspond to pairs of deletions and insertions of elements in the permutation. As a consequence, the study of translocations is closely related to the longest common subsequence problem [85] and permutation coding under the Levenshtein metric [44].

Rank modulation is by now well understood from the perspective of code construction. The capacity of rank modulation codes was derived in [36, 73, 86], while some practical code constructions were proposed in [73, 76], and further generalized in [37], [80] and [86]. Here, we complement the described work in terms of deriving upper and lower bounds on the capacity of translocation rank codes, and in terms of presenting constructive, asymptotically good coding schemes. Our constructions are based on a novel application of permutation interleaving, and are of independent interest in combinatorics and algebra. For the use of specialized forms of permutation interleaving in other areas of coding theory, the interested reader is referred to [77, 87]. Furthermore, we propose decoding algorithms for translocation codes based on decoders for codes in the Hamming metric [88, 89]. Finally, we also highlight the close relationships between permutation codes in a number of metrics.

This chapter is organized as follows. In §5.2 we provide the motivation for studying translocations as well as basic definitions used in our analysis. The

[1] There are two important motivations for increasing the number of charge levels. First, larger number of charge levels may enable storing more data, and second, when there are a large number of charge levels available, encoding methods such as push-to-the-top [80] can be used to decrease the number of times that the memory needs to be erased.

[2] Note that our definition of the term translocation differs from the definition commonly used in biology. See, e.g., [82].

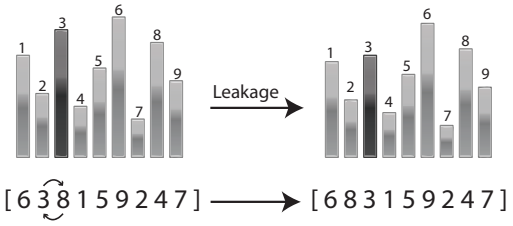


Figure 5.1: Rank modulation codes and adjacent transposition errors.

properties of permutations under translocations are studied in the same section while bounds on the size of the codes are presented in §5.3. Code constructions are presented in §5.4 and §5.5, while concluding remarks are given in §5.6.

5.2 Preliminaries

As mentioned above, our interest in translocations in permutations is motivated by rank modulation coding. In classical multi-level flash memories, each cell used for storing information is subjected to errors. As a result, classical error control schemes of non-zero rate cannot be efficiently used in such systems. One solution to the problem is to encode information in terms of rankings [9], rather than absolute values of the information sequences. Consequently, data is represented by permutations and errors manifest themselves via reordering of the ranked elements. The simplest model assumes that only elements in adjacent ranks may be exchanged. An examples is depicted in Figure 5.1.

This model has the drawback that it does not account for more general changes in ranks. With respect to this observation, consider the charge-drop model in Figure 5.2. Here, cell number 3, ranked second, experienced a leakage rate sufficiently high to move the cell's rank to the eighth position. In this framework, one translocation $\phi(i,j)$, namely $\phi(2,8)$, of length $|i-j|=6$, corresponds to 6 adjacent transpositions. Nevertheless, as already argued, a translocation should be counted as a single error, and not as a sequence of adjacent transposition errors. We also remark that translocation errors of arbitrary length accurately model *any* error that affects a single cell, and are hence suitable for modeling arbitrary charge drops of cells independently of drops of other cells, as well as read disturb and write disturb errors [39]. This makes them a good candidate for studying new error-control schemes in flash memories.

The distance arising from translocation errors, as proved in Prop. 1.2, is the Ulam distance. That is, the minimum number of translocation errors required to take a permutation π to another permutation σ equals the Ulam distance between the two permutations, or equivalently $n - \mathrm{lcs}(\pi,\sigma)$. The inequalities given in (1.5) imply that the Ulam distance is not within a constant factor from the Kendall τ distance, so that code constructions and bounds specifically

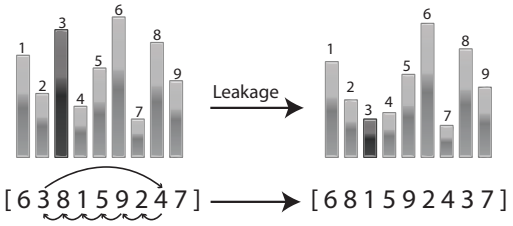


Figure 5.2: Rank modulation codes and translocation errors caused by "large" drops of charge levels.

derived for the latter distance measure are not tight and sufficiently efficient with respect to the Ulam distance.

Although the Ulam distance has received some attention in the computer science community, to the best of the authors' knowledge, codes in the Ulam distance were not reported in the literature, with the exception of [85] by Beame et al. and the notable single-error correction method by Levenshtein [44].

There exist many embedding methods for permutations, allowing one set of permutations with desirable properties according to a given distance to be mapped into another set of permutations with good properties in another metric space. In subsequent sections, we exhibit a method for interleaving permutations with good Hamming distance so as to obtain permutations with large minimum Ulam distance.

The Ulam distance and other notions introduced in this section easily extend to permutations over a set $P \subseteq [n]$. Throughout this chapter, we use the following notation and terminology in addition to the notation introduced in §1.3. For a permutation $\sigma \in \mathbb{S}_n$ and a subset P of $[n]$, the *projection* σ_P *of* σ *onto* P is obtained from σ by only keeping elements of P and removing all other elements. For example, for $\sigma = (5,4,3,2,1)$ and $P = \{2,4,5\}$, we have $\sigma_P = (5,4,2)$. Note that σ_P has length $|P|$. Furthermore, let $\mathbb{S}(P)$ stand for the set of all permutations of elements of P. The identity element of $\mathbb{S}(P)$ is e_P, obtained from $(1,2,\ldots,n)$ by removing elements that are not in P. For distinct $i,j \in P$, a translocation $\phi(i,j)$ over P is obtained from e_P by moving i to the position of j, and shifting elements between i and j, including j, by one. Right- and left-translocations over P are defined similarly.

Example 5.1. Let $P = \{2,3,5,8\}$ and consider $\pi = (5,8,3,2) \in \mathbb{S}(P)$. The translocation $\phi(8,2)$ over P equals $(8,2,3,5)$ and we have $\pi\phi(8,2) = (2,5,8,3)$. Notice that in this case, as for the case of standard permutations, the parameters in $\phi(\cdot,\cdot)$ refer to the *elements* in the corresponding identity permutation, rather than positions.

Note that a translocation may correspond to either a left- or a right-translocation. As seen from the example in Figure 5.2, right-translocations correspond to general cell leakage models. On the other hand, left-translocations assume

that the charge of a cell is increased above the level of other cells. We therefore also introduce the notion of the *right-translocation distance* in §5.2.1. As will be seen from our subsequent discussion, the Ulam distance is much easier to analyze than the right-translocation distance, and represents a natural lower bound for this distance.

5.2.1 Right-translocation Distance

Consider $\pi, \sigma \in \mathbb{S}_n$ and denote by $R_t(\pi, \sigma)$ the minimum number of right-translocations required to convert π to σ. For two permutations $\pi_1, \pi_2 \in \mathbb{S}_n$, the *right-translocation distance* $\vec{\mathsf{d}}_U(\pi_1, \pi_2)$ is defined as

$$\vec{\mathsf{d}}_U(\pi_1, \pi_2) = 2 \min_{\sigma} \max \{R_t(\pi_1, \sigma), R_t(\pi_2, \sigma)\} \ .$$

We demonstrate next that $\vec{\mathsf{d}}_U$ is in fact a metric by proving that it satisfies the triangle inequality; the other metric properties may be readily verified.

Consider three permutations, π_1, π_2, and π_3, and let

$$\sigma_{12} = \arg\min_{\sigma} \max \{R_t(\pi_1, \sigma), R_t(\pi_2, \sigma)\}$$
$$\sigma_{23} = \arg\min_{\sigma} \max \{R_t(\pi_2, \sigma), R_t(\pi_3, \sigma)\}.$$

Suppose that $\alpha_i, 1 \le i \le m_\alpha = R_t(\pi_1, \sigma_{12})$, are right-translocations and assume that $\beta_i, 1 \le i \le m_\beta = R_t(\pi_2, \sigma_{12})$, are left-translocations such that

$$\begin{aligned} \pi_1 \alpha_1 \alpha_2 \cdots \alpha_{m_\alpha} &= \sigma_{12}, \\ \sigma_{12} \beta_1 \beta_2 \cdots \beta_{m_\beta} &= \pi_2. \end{aligned} \tag{5.1}$$

Similarly, suppose that $\gamma_i, 1 \le i \le m_\gamma = R_t(\pi_2, \sigma_{23})$, are right-translocations and that $\delta_i, 1 \le i \le m_\delta = R_t(\pi_3, \sigma_{23})$, are left-translocations such that

$$\begin{aligned} \pi_2 \gamma_1 \gamma_2 \cdots \gamma_{m_\gamma} &= \sigma_{23}, \\ \sigma_{23} \delta_1 \delta_2 \cdots \delta_{m_\delta} &= \pi_3. \end{aligned} \tag{5.2}$$

Note that the existence of the sets of translocations $\{\alpha_i\}, \{\beta_i\}, \{\gamma_i\}, \{\delta_i\}$ follows from the definition of R_t.

From (5.1) and (5.2), we have

$$\pi_1 \alpha_1 \cdots \alpha_{m_\alpha} \beta_1 \cdots \beta_{m_\beta} \gamma_1 \cdots \gamma_{m_\gamma} \delta_1 \cdots \delta_{m_\delta} = \pi_3. \tag{5.3}$$

Right-translocations and left-translocations have the following simple property. Suppose β is a left-translocation and γ is a right-translocation. We can then find a right-translocation γ' and a left-translocation β' such that $\beta\gamma = \gamma'\beta'$, where either γ' or β' are allowed to be the identity permutation. Hence, (5.3)

may be rewritten as

$$\pi_1\alpha_1\cdots\alpha_{m_\alpha}\gamma'_1\cdots\gamma'_{m_\gamma}\beta'_1\cdots\beta'_{m_\beta}\delta_1\cdots\delta_{m_\delta} = \pi_3. \tag{5.4}$$

Next, let $\sigma_{13} = \pi_1\alpha_1\cdots\alpha_{m_\alpha}\gamma'_1\cdots\gamma'_{m_\gamma}$. Note that σ_{13} is not required to be the minimizer of $\max\{R_t(\pi_1,\sigma), R_t(\pi_3,\sigma)\}$.

From (5.4) and the fact that α_i and γ'_i are right-translocations and β'_i and δ_i are left-translocations, it follows that

$$\begin{aligned} R_t(\pi_1,\sigma_{13}) &\le m_\alpha + m_\gamma = R_t(\pi_1,\sigma_{12}) + R_t(\pi_2,\sigma_{23}), \\ R_t(\pi_3,\sigma_{13}) &\le m_\beta + m_\delta = R_t(\pi_2,\sigma_{12}) + R_t(\pi_3,\sigma_{23}), \end{aligned}$$

and thus

$$\begin{aligned} \vec{\mathsf{d}}_U(\pi_1,\pi_3) &\le 2\max\{R_t(\pi_1,\sigma_{13}), R_t(\pi_3,\sigma_{13})\} \\ &\le 2\max\big\{R_t(\pi_1,\sigma_{12}) + R_t(\pi_2,\sigma_{23}), \\ &\qquad R_t(\pi_2,\sigma_{12}) + R_t(\pi_3,\sigma_{23})\big\} \\ &\le 2\max\{R_t(\pi_1,\sigma_{12}), R_t(\pi_2,\sigma_{12})\} + \\ &\qquad 2\max\{R_t(\pi_2,\sigma_{23}), R_t(\pi_3,\sigma_{23})\} \\ &= \vec{\mathsf{d}}_U(\pi_1,\pi_2) + \vec{\mathsf{d}}_U(\pi_2,\pi_3). \end{aligned}$$

Hence, $\vec{\mathsf{d}}_U$ satisfies the triangle inequality.

The definition of $\vec{\mathsf{d}}_U$ implies that for two permutations $\pi_1, \pi_2 \in \mathbb{S}_n$, one has $\vec{\mathsf{d}}_U(\pi_1,\pi_2) \le 2t$ if and only if there exists a permutation $\sigma \in \mathbb{S}_n$ such that $R_t(\pi_1,\sigma) \le t$ and $R_t(\pi_2,\sigma) \le t$. Hence, a code C is t-right-translocation correcting if and only if $\vec{\mathsf{d}}_U(\pi_1,\pi_2) > 2t$ for all $\pi_1,\pi_2 \in C$, $\pi_1 \ne \pi_2$. This means that under the given distance constraint, it is not possible to confuse the actual codeword π_1 with another (wrong) codeword π_2.

Observe that the following bound holds:

$$\mathsf{d}_U(\pi_1,\pi_2) \le \vec{\mathsf{d}}_U(\pi_1,\pi_2).$$

It is straightforward to characterize the minimum number of right-translocations needed to transform one permutation to another, as we show below.

Definition 5.2. Let $\pi,\sigma \in \mathbb{S}_n$. We denote the set of π^{-1}-increasing and σ^{-1}-decreasing elements as

$$\begin{aligned} J(\pi,\sigma) := \big\{ & i \in [n] : \exists j, \pi^{-1}(i) < \pi^{-1}(j) \\ & \text{and } \sigma^{-1}(i) > \sigma^{-1}(j)\big\}. \end{aligned}$$

Note that $|J|$ is left-invariant since, for $\pi, \sigma, \omega \in \mathbb{S}_n$,

$$\begin{aligned}
|J(\omega\pi, \omega\sigma)| &= \left|\left\{i \in [n] : \exists j, \pi^{-1}\omega^{-1}(i) < \pi^{-1}\omega^{-1}(j)\right.\right. \\
&\qquad \left.\left. \text{and } \sigma^{-1}\omega^{-1}(i) > \sigma^{-1}\omega^{-1}(j)\right\}\right| \\
&= \left|\left\{i' \in [n] : \exists j', \pi^{-1}(i') < \pi^{-1}(j')\right.\right. \\
&\qquad \left.\left. \text{and } \sigma^{-1}(i') > \sigma^{-1}(j')\right\}\right| \\
&= |J(\pi, \sigma)|,
\end{aligned}$$

where for the first equality we have used the fact that $(\omega\pi)^{-1} = \pi^{-1}\omega^{-1}$ and $(\omega\sigma)^{-1} = \sigma^{-1}\omega^{-1}$, and the second equality can be obtained by letting $i' = \omega^{-1}(i)$ and $j' = \omega^{-1}(j)$. Furthermore, it is not difficult to show that R_t is left-invariant.

Lemma 5.3. *Let $\pi, \sigma \in \mathbb{S}_n$. Then*

$$R_t(\pi, \sigma) = |J(\pi, \sigma)| .$$

Proof. It suffices to show that

$$R_t(\pi, e) = |J(\pi, e)| ,$$

where

$$|J(\pi, e)| = \left|\left\{i \in [n] : \exists j < i, \pi^{-1}(i) < \pi^{-1}(j)\right\}\right| .$$

Let π_1 be obtained from π by applying a right-translocation that moves some element k to the right. Every element of $J(\pi, e)\backslash\{k\}$ is also in $J(\pi_1, e)$ as each element of $J(\pi, e)\backslash\{k\}$ is involved in at least one inversion, which is not affected by moving k. Hence, $|J(\pi_1, e)| \geq |J(\pi, e)| - 1$ with equality if $J(\pi_1, e) = J(\pi, e)\backslash\{k\}$. Repeating the same argument yields $R_t(\pi, e) \geq |J(\pi, e)| - |J(e, e)| = |J(\pi, e)|$.

Conversely, to transform π into e, it suffices to apply to each $i \in J$ the shortest right-translocation that moves this element to the smallest position i', such that to the left of position i' are all elements smaller than i. Hence, $R_t(\pi, e) \leq |J(\pi, e)|$. □

For permutations $\pi, \sigma \in \mathbb{S}_n$, the difference between $R_t(\pi, \sigma)$ and $\mathsf{d}_U(\pi, \sigma)$ may be as large as $n - 2$. This may be seen by letting $\pi = (2, 3, \ldots, n, 1)$ and $\sigma = e$, and observing that $R_t(\pi, \sigma) = n - 1$ and $\mathsf{d}_U(\pi, \sigma) = 1$. Furthermore, it can be shown that this is the largest possible gap. To prove this fact, first note that $R_t(\pi, \sigma) = 0$ if and only if $\mathsf{d}_U(\pi, \sigma) = 0$ and thus to obtain a positive gap, one must have $\mathsf{d}_U(\pi, \sigma) \geq 1$. We also have $\mathsf{d}_U(\pi, \sigma) \leq R_t(\pi, \sigma) \leq n - 1$. Hence, $1 \leq \mathsf{d}_U(\pi, \sigma) \leq R_t(\pi, \sigma) \leq n - 1$, which implies that the gap is at most $n - 1 - 1 = n - 2$.

5.3 Bounds on the Size of Codes

5.3.1 Codes in the Ulam Distance

Henceforth, a *permutation code*, or simply a code, of length n and minimum distance d in a metric d refers to a subset C of $\mathbb{S}_n$ such that for all distinct $\pi, \sigma \in C$, we have $\mathsf{d}(\pi, \sigma) \geq d$.

Let $A_\circ(n, d)$ be the maximum size of a permutation code of length n and minimum Ulam distance d.

Proposition 5.4. *For all integers n and d with $n \geq d \geq 1$, we have*

$$A_\circ(n, d) \geq \frac{(n-d+1)!}{\binom{n}{d-1}}.$$

Proof. Let $B_\circ(r)$ be the number of permutations at Ulam distance at most r from a given permutation. From left-invariance, we have

$$B_\circ(r) = |\{\sigma : \mathsf{d}_U(\sigma, e) \leq r\}|.$$

The permutations that are within Ulam distance r from e are precisely the permutations σ with $\mathrm{lcs}(e, \sigma) \geq n - r$. There are $\binom{n}{r}$ ways to choose the first $n-r$ elements of the longest common subsequence of e and σ and at most $\frac{n!}{(n-r)!}$ ways to arrange the remaining elements of σ. Hence,

$$B_\circ(r) \leq \binom{n}{r} \frac{n!}{(n-r)!}.$$

From the Gilbert-Varshamov bound, we have $A_\circ(n, d) \geq \frac{n!}{B_\circ(d-1)}$ and thus

$$A_\circ(n, d) \geq \frac{n!}{\binom{n}{d-1} \frac{n!}{(n-d+1)!}},$$

which completes the proof. □

Proposition 5.5. *For all $n, d \in \mathbb{Z}$ with $n \geq d \geq 1$,*

$$A_\circ(n, d) \leq (n-d+1)!.$$

Proof. We provide two proofs for this bound. The first proof is based on a projection argument first described in [36], while the second proof is based on a standard counting argument.

1. Let C be a code of length n, size M, and minimum distance d. Let k be the smallest integer such that $\sigma_{\{1,\ldots,k+1\}} \neq \pi_{\{1,\ldots,k+1\}}$ for all distinct $\sigma, \pi \in C$. Hence, $M \leq (k+1)!$. By definition, there exist $\sigma_1, \sigma_2 \in C$ such that $\sigma_{\{1,\ldots,k\}} = \pi_{\{1,\ldots,k\}}$. So, $\mathrm{lcs}(\sigma_1, \sigma_2) \geq k$ and thus $d \leq \mathsf{d}_U(\sigma_1, \sigma_2) \leq n-k$. Hence, $M \leq (n-d+1)!$.

2. Again, let C be a code of length n, size M, and minimum distance d. Since the minimum distance is d, all $M\binom{n}{n-d+1}$ subsequences of length $n-d+1$ of the codewords of C are unique. There are $\frac{n!}{(d-1)!}$ possible subsequences of length $n-d+1$. Hence,

$$M\binom{n}{n-d+1} \leq \frac{n!}{(d-1)!}$$

which implies that $M \leq (n-d+1)!$.

□

From the two previous propositions, we obtain

$$\frac{(n-d+1)!}{\binom{n}{d-1}} \leq A_\circ(n,d) \leq (n-d+1)! \tag{5.5}$$

In what follows, all limits are evaluated for $n \to \infty$, unless stated otherwise.

Lemma 5.6. *The following results hold:*

1. $$\lim \frac{\ln(n-d(n))!}{\ln n!} = 1 - \lim \frac{d(n)}{n},$$

2. $$\lim \frac{\ln \frac{n!}{d(n)!}}{\ln n!} = 1 - \lim \frac{d(n)}{n},$$

3. $$\lim \frac{\ln \binom{n}{d(n)}}{\ln n!} = 0.$$

Proof. All claims follow easily from the asymptotic formula $\ln(n!) = n\ln n + O(n)$. □

Let $\mathcal{C}_\circ(d)$ denote the capacity of translocation codes of minimum Ulam distance d, i.e., $\mathcal{C}_\circ(d) = \lim_{n\to\infty} \frac{\ln A_\circ(n,d)}{\ln n!}$. *A capacity achieving code* refers to a code with maximum rate and a given minimum distance, in a given metric space.

Theorem 5.7. *The capacity of translocation codes of minimum distance $d = d(n)$ equals $\mathcal{C}_\circ(d) = 1-\delta$, where $\delta = \lim \frac{d(n)}{n}$.*

Proof. From (5.5), we have

$$\begin{aligned} \frac{\ln(n-d+1)! - \ln\binom{n}{d-1}}{\ln n!} &\leq \frac{\ln A_\circ(n,d)}{\ln n!} \\ &\leq \frac{\ln(n-d+1)!}{\ln n!} \end{aligned} \tag{5.6}$$

Taking the limit of (5.6) and using Lemma 5.6 proves the theorem. □

At this point, it is worth observing that the problem of bounding the longest common subsequence in permutations has been recently studied in a combinatorial framework [85]. There, the question of interest was to determine the minimum length of the longest common subsequence between any two distinct permutations from a set of k permutations of length n. When translated into the terminology of translocation codes, the problem reduces to finding $d_k(n)$, the largest possible minimum Ulam distance of a set of k permutations of $\mathbb{S}_n$.

The bounds derived in [85] are constructive, but they hold only in the *zero-capacity domain of the code parameters*. A more detailed description of one of the constructions of [85] is presented in §5.5.5. The bounds of [85] imply that $d_k(n) \geq n - 32(nk)^{1/3}$ for $3 \leq k \leq \sqrt{n}$. Hence, for $n - 32\sqrt{n} \leq d \leq n - 32(3n)^{1/3}$,

$$A_\circ(n,d) \geq \frac{1}{n}\left(\frac{n-d}{32}\right)^3.$$

Furthermore, for $k \geq 4$, $d_k(n) \geq n - \left\lceil n^{1/(k-1)} \right\rceil^{k/2-1}$. For $k \geq 2(1+\log_2 n)$, this bound is of no practical use.

For $1 + \log_2 n \leq k < 2(1+\log_2 n)$, one has $d_k(n) \geq n - 2^{k/2-1}$, which implies that

$$A_\circ(n,d) \geq 2(1+\log_2(n-d))$$

for $d \leq n - \sqrt{n/2}$. Similar bounds can be obtained for $A_\circ(n,d)$ by assuming that $m - 1 < n^{\frac{1}{k-1}} \leq m$ for some integer $m \leq \left\lceil n^{1/3} \right\rceil$. Note that although these results hold for the zero-capacity regime, they still may be useful for finite codelength analysis.

Remark: Similar bounds may be derived for the asymmetric regime of translocation error-correcting codes. For this purpose, let

$$B'(r) = \left|\{\sigma : R_t(e,\sigma) \leq r\}\right|,$$
$$\vec{B}(r) = \left|\left\{\sigma : \vec{\mathsf{d}}_U(e,\sigma) \leq r\right\}\right|.$$

Then

$$\frac{n!}{B_\circ(2t)} \leq \frac{n!}{\vec{B}(2t)} \leq \vec{A}(n,2t+1) \leq \frac{n!}{B'(t)},$$

where $\vec{A}(n,d)$ denotes the maximum size of a permutation code with minimum right-translocation distance d.

5.3.2 Codes in Other Metrics

Translocation errors, and consequently, translocation error correcting codes, are difficult to analyze directly. On the other hand, as described in §1.3.4.5, the Ulam distance is related to various other metrics well-studied in the coding theory and mathematics literature. Since the constructions in subsequent sections rely on codes in other metrics on permutations, we provide a brief overview of

the state of the art results pertaining to the Hamming, Cayley, and Kendall τ metrics. We also supplement the known findings with a number of new results for the metrics under consideration.

5.3.2.1 Hamming Metric

Codes in the Hamming metric have a long history, dating back to the work [69]. The Hamming metric is a suitable distance measure for use in power line communication systems [19].

Let $A_H(n,d)$ denote the largest number of permutations of length n and minimum Hamming distance d. Frankl and Deza [90, Theorem 4] and Deza [91] showed that

$$\frac{n!}{B_H(d-1)} \leq A_H(n,d) \leq \frac{n!}{(d-1)!},$$

where $B_H(r)$ is the volume of the sphere of radius r in the space of permutations with Hamming metric. Improvements of these results for some special cases were also obtained via linear programing methods; see for example [92].

Let D_i denote the number of derangements of i objects, i.e., the number of permutations of $[i]$ at Hamming distance i from the identity permutation. It can be shown that $B_H(r) = 1 + \sum_{i=2}^{r} \binom{n}{i} D_i$. Hence,

$$\begin{aligned} B_H(d-1) &= 1 + \sum_{i=2}^{d-1} \binom{n}{i} D_i \leq \sum_{i=1}^{d-1} \frac{n!}{(n-i)!} \\ &\leq (d-1)\frac{n!}{(n-d+1)!} \end{aligned}$$

where the first inequality follows from the fact that $D_i \leq i!$. Note that although a more precise asymptotic characterization for the number of derangements is known, namely

$$\lim_{\ell \to \infty} \frac{D_\ell}{\ell!} = \frac{1}{e},$$

the simple bound $D_i \leq i!$ is sufficiently tight for the capacity computation.

The above results lead to

$$\frac{(n-d+1)!}{d-1} \leq A_H(n,d) \leq \frac{n!}{(d-1)!}.$$

Let $\mathcal{C}_H(d)$ denote the capacity of permutation codes with minimum Hamming distance d, i.e., $\mathcal{C}_H(d) = \lim \frac{\ln A_H(n,d)}{\ln n!}$. Lemma 5.6 implies the following theorem.

Theorem 5.8. $\mathcal{C}_H = 1 - \lim \frac{d(n)}{n}$.

5.3.2.2 Cayley Metric

Let $A_T(n,d)$ denote the maximum size of a code with minimum Cayley distance d_T at least d. From (1.2), we have

$$A_H(n,2d) \le A_T(n,d) \le A_H(n,d).$$

Using the bounds above, we have the following theorem regarding the capacity $\mathcal{C}_T(d)$ of permutation codes of minimum distance d in the Cayley metric.

Theorem 5.9. *The capacity of permutation codes of distance d in the Cayley metric is bounded as*

$$1 - 2\lim \frac{d(n)}{n} \le \mathcal{C}_T(d) \le 1 - \lim \frac{d(n)}{n}.$$

5.3.2.3 Kendall τ Metric

Let $A_K(n,d)$ denote the largest cardinality of a permutation code of length n with minimum Kendall τ distance d, and let $\mathcal{C}_K(d) = \lim \frac{\ln A_K(n,d)}{\ln n!}$. Barg and Mazumdar [36, Theorem 3.1] showed that

$$\mathcal{C}_K(d) = 1 - \epsilon, \qquad \text{for } d = \Theta\left(n^{1+\epsilon}\right).$$

Note that for the Kendall τ, the maximum distance between two permutations may be as large as $\Theta(n^2)$. On the other hand, the diameter of $\mathbb{S}_n$ with respect to the Ulam distance is $\Theta(n)$.

5.3.2.4 Levenshtein Metric

The bounds on the size of deletion/insertion correcting codes in the more general case of codes with distinct symbols were first derived by Levenshtein in his landmark paper [44]. The lower bound relies on the use of Steiner triple systems and designs [44]. More precisely, let $D(n,q)$ be the largest cardinality of a set of n-subsets of the set $\{0,1,\ldots,q-1\}$ with the property that every $(n-1)$-subset of $\{0,1,\ldots,q-1\}$ is a subset of at most one of the n-subsets. Then the following result holds for the cardinality $\mathcal{A}_L(n,q)$ of the largest single-deletion correcting codes consisting of codewords in $\{0,1,\ldots,q-1\}^n$ with distinct symbols [44]:

$$(n-1)!D(n,q) \le \mathcal{A}_L(n,q) \le \frac{q!}{n\,(q-n+1)!}.$$

5.4 Single-Error Correcting Codes

This section contains constructions for single-translocation error detecting and single-translocation error correcting codes. For the latter case, we exhibit two

constructions, one for translocations and another for right-translocations.

5.4.1 Detecting a Single Translocation Error

We start by describing a code that can detect a single translocation error. From the discussion in §5.2, recall that the Ulam distance is half of the Levenshtein distance and thus any single-deletion correcting code may be used for detecting a single translocation error. An elegant construction for single-deletion correcting permutation codes was described by Levenshtein in [44]. The resulting code has cardinality $(n-1)!$ and is optimal since, from Proposition 5.5, we have

$$A_\circ(n,2) \le (n-2+1)! = (n-1)!.$$

Hence, $A_\circ(n,2) = (n-1)!$.

Levenshtein's construction is of the following form.

Let

$$W_2^n = \left\{ u \in \{0,1\}^n : (n+1) \mid \sum_{i=1}^{n} iu(i) \right\}$$

where $a \mid b$ denotes that a is a divisor of b. For $\sigma \in \mathbb{S}_n$, let

$$Z(\sigma) = (z(1), \ldots, z(n-1))$$

be a vector with

$$z(i) = \begin{cases} 0, & \text{if } \sigma(i) \le \sigma(i+1), \\ 1, & \text{if } \sigma(i) > \sigma(i+1), \end{cases} \qquad i \in [n-1].$$

The code

$$C = \left\{ \sigma \in \mathbb{S}_n : Z(\sigma) \in W_2^{n-1} \right\} \tag{5.7}$$

of size $(n-1)!$ is capable of correcting a single deletion. Hence, this code can detect a single translocation error.

Let $\mathbb{S}_n^{(t)}$ be the set of sequences σ of length $n-t$ that can be obtained from some permutation in $\mathbb{S}_n$ by t deletions. In other words, $\mathbb{S}_n^{(t)}$ is the set of words of length $n-t$ from the alphabet $[n]$ without repetitions. A code $C_p \subseteq \mathbb{S}_n$ is a *perfect* code capable of correcting t deletions if, for every $\sigma \in \mathbb{S}_n^{(t)}$, there exists a unique $\pi \in C_p$ such that σ can be obtained from π by t deletions. It was shown in [44] that C in (5.7) is a perfect code capable of correcting a single deletion.

The minimum Levenshtein distance of C_p is $2(t+1)$ and thus the minimum Ulam distance of C_p is $t+1$. Since the size of $\mathbb{S}_n^{(t)}$ equals $\binom{n}{n-t}(n-t)!$ and $\binom{n}{t}$ elements of $\mathbb{S}_n^{(t)}$ can be obtained by t deletions from each $\sigma \in C_p$, we have that $|C_p| = (n-t)!$. Recall from Prop. 5.5 that the size of a code with minimum Ulam distance $t+1$ is $\le (n-t)!$. Thus a perfect code capable of correcting t deletions, if it exists, is a rate-optimal code in the Ulam metric. Although conditions for the

existence of such codes were investigated in [44], both necessary and sufficient conditions are known only for a small number of special cases.

In the next two subsections, we describe codes capable of correcting a single right-translocation error and codes capable of correcting a single translocation error. In the constructions, we make use of a single-transposition error detecting code, described below.

A single-transposition error detecting code

Applying a transposition to a permutation changes the parity of the permutation. Also, for $n \geq 2$, half of the permutations in $\mathbb{S}_n$ are even and half of them are odd.[3] Hence, the code C containing all even permutations of $\mathbb{S}_n$ is a single-transposition error detecting code of length n and cardinality $n!/2$.

5.4.2 Correcting a Single Right-translocation Error

Next, we present a construction for codes that correct a single right-translocation error. For this purpose, we first define the operation of permutation interleaving and the operation of code interleaving.

Definition 5.10. For vectors $\sigma_i, i \in [k]$, of lengths m_i with $m_1 \geq m_2 \geq \cdots \geq m_k \geq m_1 - 1$, the *interleaved vector* $\sigma = \sigma_1 \circ \sigma_2 \circ \cdots \circ \sigma_k$ is obtained by alternatively placing the elements of $\sigma_1, \sigma_2, \ldots, \sigma_k$ in order. That is,

$$\sigma(j) = \sigma_i(\lceil j/k \rceil), \qquad 1 \leq j \leq \sum_{i=1}^{k} m_i \tag{5.8}$$

where $i \equiv j \pmod{k}$. For a class of k codes $C_i, i \in [k]$, let

$$C_1 \circ \cdots \circ C_k = \{\sigma_1 \circ \cdots \circ \sigma_k : \sigma_i \in C_i, i \in [k]\}. \tag{5.9}$$

For example, for vectors σ and π of length m, we have

$$\sigma \circ \pi = (\sigma(1), \pi(1), \sigma(2), \pi(2), \ldots, \sigma(m), \pi(m)) \tag{5.10}$$

and for vectors σ and π of lengths m and $m-1$ respectively, we have

$$\sigma \circ \pi = (\sigma(1), \pi(1), \sigma(2), \pi(2), \ldots, \sigma(m)). \tag{5.11}$$

The following proposition introduces codes that can correct a single right-translocation error. The decoding algorithm is contained in the proof of the proposition.

[3] More precisely, the symmetric group can be partitioned into the alternating group and its coset.

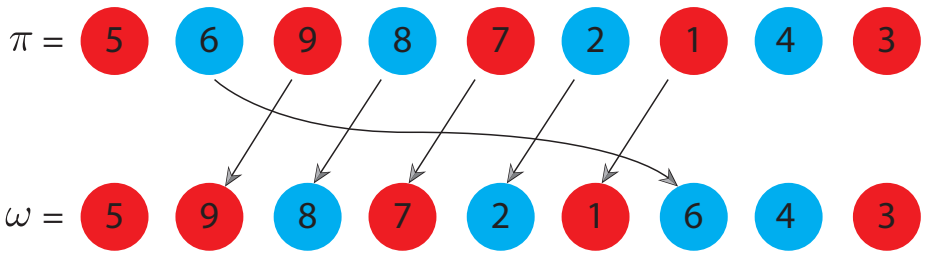


Figure 5.3: The effect of the right-translocation error $\phi(2,7)$ on the codeword $\pi = (5,6,9,8,7,2,1,4,3)$. The result is the word $\omega = (5,9,8,7,2,1,6,4,3)$.

Proposition 5.11. *Let $P_i, i = 1,2$, be the set of odd and even numbers in $[n]$, respectively, and let C_i be the set of even permutations of P_i for $i = 1,2$. The interleaved code $C = C_1 \circ C_2$ corrects a single right-translocation error.*

Proof. Given the permutation $\omega \notin C$, we want to find the unique $\pi \in C$ such that $\omega = \pi\phi(i,j)$, $i < j$. An example is shown in Figure 5.3, with

$$\omega = (5,9,8,7,2,1,6,4,3)$$

and π unknown to the decoder.

The kth element of ω is out of place if $k \not\equiv \omega(k) \pmod 2$. It is easy to see that

$$i = k_{\min} := \min\{k : k \not\equiv \omega(k) \pmod 2\},$$

i.e., i equals the smallest integer k such that the kth element of ω is out of place. In the example shown in Figure 5.3, $i = 2$. Finding j is slightly more complicated since we must consider two different cases depending on the parity of the length $j - i$ of the right-translocation.

Let $k_{\max} := \max\{k : k \not\equiv \omega(k) \pmod 2\}$. If $j - i$ is odd, then $j = k_{\max}$. Otherwise, $j = k_{\max} + 1$. That is, the right-translocation error is either $\phi(i, k_{\max})$ or $\phi(i, k_{\max} + 1)$. Thus, the codeword π either equals $\pi' = \omega\phi(k_{\max}, i)$ or equals $\pi'' = \omega\phi(k_{\max} + 1, i)$. In the example of Figure 5.3, we have

$$\pi' = (5,6,9,8,7,2,1,4,3),$$
$$\pi'' = (5,4,9,8,7,2,1,6,3).$$

To find which of the two cases is correct, we proceed as follows.

Since $\omega = \pi''\phi(i, k_{\max} + 1)$ and $\pi' = \omega\phi(k_{\max}, i)$, we have

$$\begin{aligned}\pi' &= \pi''\phi(i, k_{\max} + 1)\phi(k_{\max}, i) \\ &= \pi''\langle i\ k_{\max} + 1\rangle.\end{aligned}$$

Recall that if $j - i$ is odd, then $j = k_{\max}$, and if $j - i$ is even, then $j = k_{\max} + 1$. Hence, $i \equiv k_{\max} + 1 \pmod 2$. In both π' and π'', the parity of the elements is

the same as the parity of their positions. Thus, the transposition $\langle i\ k_{\max}+1\rangle$ affects only elements of the same parity as i. Hence, if i is odd, then $\pi'_{P_2} = \pi''_{P_2} = \omega_{P_2} = \pi_{P_2}$ and if i is even, then $\pi'_{P_1} = \pi''_{P_1} = \omega_{P_1} = \pi_{P_1}$.

Without loss of generality, assume that i is even. Then $\pi'_{P_1} = \pi''_{P_1} = \omega_{P_1} = \pi_{P_1}$ and the subwords π'_{P_2} and π''_{P_2} differ in one transposition. Since C_2 has minimum Cayley distance two, only one of π'_{P_2} and π''_{P_2} belongs to C_2, and so π can be uniquely determined as being either equal to π' or π''. □

The cardinality of the interleaved code $C = C_1 \circ C_2$ equals $\frac{1}{4}\lceil \frac{n}{2}\rceil!\lfloor \frac{n}{2}\rfloor!$, and its rate asymptotically equals

$$\frac{\ln(1/4) + 2\ln\lfloor \frac{n}{2}\rfloor!}{\ln n!} \sim \frac{n\ln n + O(n)}{n\ln n + O(n)} \sim 1.$$

5.4.3 Correcting a Single Translocation Error

The construction of the previous subsection can be extended to generate codes capable of correcting a single translocation error as stated in the following proposition. Although the proposition is stated for n being a multiple of three, it can be easily extended to other cases.

Proposition 5.12. *Suppose n is a multiple of three. Let $P_i, i=1,2,3$, be the set of numbers in $[n]$ that are equal to i modulo three, and let C_i be the set of even permutations of P_i for $i = 1,2,3$. The interleaved code $C = C_1 \circ C_2 \circ C_3$ corrects a single translocation error.*

Proof. Suppose that π is the stored permutation, ω is the retrieved permutation, and that the error is the translocation $\phi(i,j)$. If $|i-j| = 1$, then $\phi(i,j)$ can be easily identified. Suppose that $|i-j| > 1$. The translocation $\phi(i,j)$ moves $|i-j|$ elements of π one position to the left, provided that $i < j$, or one position to the right, provided that $j < i$. In either case, one element moves in the "opposite direction" from the other elements. Hence, for $|i-j| > 1$ the direction of the translocation (left or right) can be identified.

Once the direction of the translocation is known, i can be found as follows: if the error is a right-translocation, then

$$i = \min\{k : \omega(k) \not\equiv k \pmod 3\},$$

and if the error is a left-translocation, then

$$i = \max\{k : \omega(k) \not\equiv k \pmod 3\}.$$

For simplicity, suppose the error is a right-translocation. The proof for left-translocations is similar. Let $k_{\max} \coloneqq \max\{k : k \not\equiv \omega(k) \pmod 3\}$. We have the

following three cases. If $j - i \equiv 0 \pmod 3$, then $j = k_{\max} + 1$ and

$$\omega(k_{\max}) \equiv \omega(k_{\max} + 1) \pmod 3.$$

If $j - i \equiv 1 \pmod 3$, then $j = k_{\max}$ and

$$\omega(k_{\max}) \not\equiv \omega(k_{\max} + 1) \pmod 3.$$

Finally, if $j - i \equiv 2 \pmod 3$, then $j = k_{\max}$ and

$$\omega(k_{\max}) \equiv \omega(k_{\max} + 1) \pmod 3.$$

So, if $\omega(k_{\max}) \not\equiv \omega(k_{\max} + 1) \pmod 3$, then $j = k_{\max}$ and π is uniquely determined as $\omega\phi(k_{\max}, i)$. Otherwise, the error is either $\phi(i, k_{\max})$ or $\phi(i, k_{\max} + 1)$. Let $\pi' = \omega\phi(k_{\max}, i)$ and $\pi' = \omega\phi(k_{\max} + 1, i)$. Then, $\pi' = \pi'' \langle i \ k_{\max} + 1 \rangle$ and similar to the proof of Prop. 5.11, it can be shown that π' and π'' are not both in C. Hence, π can be determined as either being equal to π' or π''. □

Example 5.13. Consider the single-translocation correcting code for $n = 12$. For this case, we have $P_1 = \{1, 4, 7, 10\}, P_2 = \{2, 5, 8, 11\}$, and $P_3 = \{3, 6, 9, 12\}$.

Suppose that the stored codeword is π, the error is $\phi(i, j)$, and the retrieved word is

$$\omega = (1, 6, 10, 8, 3, 7, 5, 11, 12, 4, 2, 9).$$

Given ω, the decoder first identifies the elements that are out of order, i.e., elements that are not equivalent to their positions modulo three – in this case, $\{6, 10, 8, 3, 7, 5\}$. Since more than two elements are out of order, we have $|i-j| > 1$. Furthermore, since more than two elements have moved one position to the left, ϕ is a right-translocation. Observe that $k_{\max} = 7$ and that $\omega(k_{\max}) \equiv \omega(k_{\max} + 1) \pmod 3$. Hence, we let

$$\pi' = \omega\phi(7, 2) = (1, 5, 10, 8, 7, 11, 4, 2)$$
$$\pi'' = \omega\phi(8, 2) = (1, 11, 10, 8, 7, 5, 4, 2).$$

We then have $\pi'_{P_2} = (5, 8, 11, 2)$ and $\pi''_{P_2} = (11, 8, 5, 2)$. Since only π''_{P_2} is an even permutation, the error is $\phi(2, 8)$ and thus $\pi = (1, 11, 6, 10, 8, 3, 7, 5, 12, 4, 2, 9)$.

The cardinality of the code equals $\left(\frac{1}{2}\left(\frac{n}{3}\right)!\right)^3$, while its rate equals

$$\frac{3 \ln(1/2) + 3 \ln\left(\frac{n}{3}\right)!}{\ln n!} \sim \frac{n \ln n + O(n)}{n \ln n + O(n)} \sim 1.$$

5.5 t-translocation Error Correcting Codes

We describe next some general constructions for t-translocation error-correcting codes, starting with an extension of the interleaving methods from §5.4.

5.5.1 Interleaving Codes in Hamming Metric

We construct a family of codes with Ulam distance $2t+1$, length $n = s(2t+1)$ for some integer $s \geq 4t+1$, and cardinality $M = (A_H(s, 4t+1))^{2t+1}$, where $A_H(s,d)$, as before, denotes the maximum size of a permutation code with length s and minimum Hamming distance d. The construction relies on the use of $2t+1$ permutation codes, each with minimum Hamming distance at least $4t+1$. First, we present the proposed construction and then prove that the minimum Ulam distance of the code is at least $2t+1$.

For a given n and t, where $n \equiv 0 \pmod{2t+1}$, partition the set $[n]$ into $2t+1$ classes P_i, each of size s, with

$$P_i = \{j \in [n] : j \equiv i \pmod{2t+1}\}, \qquad i \in [2t+1]. \tag{5.12}$$

For example, for $t = 2$ and $n = 45$, one has

$$P_1 = \{1, 6, \ldots, 41\},$$
$$P_2 = \{2, 7, \ldots, 42\},$$
$$P_3 = \{3, 8, \ldots, 43\},$$

and so on.

For $i \in [2t+1]$, let C_i be a permutation code over P_i with minimum Hamming distance at least $4t+1$.[4] The code C is obtained by interleaving the codes C_i, i.e., $C = C_1 \circ \cdots \circ C_{2t+1}$, and is referred to as an interleaved code with $2t+1$ classes. In the interleaved code, the s elements of P_i occupy positions that are equivalent to i modulo $2t+1$.

The following theorem provides a lower-bound for the minimum Ulam distance of C. The proof of the theorem is presented after stating the required definitions, and three technical lemmas.

Theorem 5.14. *Assume we are given three positive integers s, t, $n = s(2t+1)$, and a partition of $[n]$ of the form given in (5.12). If, for $i \in [2t+1]$, C_i is a permutation code over P_i with minimum Hamming distance at least $4t+1$, then $C = C_1 \circ \cdots \circ C_{2t+1}$ is a permutation code over $[n]$ with minimum Ulam distance greater than or equal to $2t+1$.*

Corollary 5.15. *For the code C of Theorem 5.14 and distinct $\sigma, \pi \in C$, the length of the longest common subsequence of π and σ is less than $n - 2t$.*

For convenience, we introduce an alternative notation for translocations. Let the mapping $\psi : \mathbb{S}_n \to \mathbb{S}_n$ be defined as follows. For a permutation $\sigma \in \mathbb{S}_n$, an integer ℓ, and $a \in [n]$, let $\psi(a, \ell)\sigma$ denote the permutation obtained from σ by moving the element a exactly $|\ell|$ positions to the right if $\ell \geq 0$ and to the left if

[4] It is clear that instead of using permutation codes for interleaving, one can also use codes with distinct symbols such as those described in [93].

$\ell \le 0$. In other words, for any permutation $\sigma \in \mathbb{S}_n$ and $a \in [n]$,

$$\psi(a,\ell)\sigma = \sigma\phi\big(\sigma^{-1}(a), \sigma^{-1}(a)+\ell\big).$$

For example, we have $\psi(4,3)(3,4,2,5,1) = (3,2,5,1,4) = (3,4,2,5,1)\phi(2,5)$. Note that the mapping ψ is written multiplicatively. Furthermore, with slight abuse of terminology, $\psi(a,\ell)$ may also be called a translocation.

Consider $\sigma, \pi \in \mathbb{S}_n$ with distance $\mathtt{d}_U(\sigma,\pi) = m$. A *$\psi$-transform* converting σ to π is a sequence $\psi_1, \psi_2, \ldots, \psi_m$ of translocations such that $\pi = \psi_m \cdots \psi_2 \psi_1 \sigma$.

Let $b_1 < b_2 < \cdots < b_m$ be the elements of $[n]$ that are not in the longest common subsequence of σ and π. Each b_k is called a *displaced* element. The set $\{b_1, \ldots, b_m\}$ is called the *set of displaced elements* and is denoted by $D(\sigma,\pi)$.

The *canonical* transform converting σ to π is a ψ-transform $\psi_1, \psi_2, \ldots, \psi_m$ with $\psi_k = \psi(b_k, \ell_k)$ for appropriate choices of $\ell_k, k \in [m]$. In other words, the canonical transform operates only on displaced elements and corresponds to the shortest sequence of translocations that convert σ to π.

As an example, consider

$$\sigma = (1,2,3,4,5,6,7,\ldots,15),$$
$$\pi = (1,3,4,5,6,2,7,\ldots,15).$$

Here, $n = 15$, $t = 1$, and $s = 3$. The canonical transform is $\psi(2,4)$ and we have

$$\pi = \psi(2,4)\sigma = (1,3,4,5,6,2,7,\ldots,15). \tag{5.13}$$

In this example, $D(\sigma,\pi) = \{2\}$.

Let $\pi_l = \psi_l \cdots \psi_2 \psi_1 \sigma$ for $1 \le l \le m$. An element a is *moved over* an element b if a is on one side of b in π_{j-1} and on the other side of b in π_j for some $j \in [m]$. In the above example with $\psi(2,4)$, 2 is moved over 4 but it is not moved over 7.

An element $k \in [2t+1]$ is called a *σ,π−pivot*, or simply a pivot, if no element of P_k is displaced, i.e., $P_k \cap D(\sigma,\pi) = \varnothing$. In the example corresponding to (5.13), the pivots are 1 and 3.

For $I \subseteq [2t+1]$, define P_I as $\cup_{i\in I} P_i$. Also, recall that for $\omega \in \mathbb{S}_n$ and a set P, ω_P denotes the projection of ω onto P. For example, for $t = 1$, $n = 15$, $\omega = (1,2,3,\ldots,15)$, and $I = \{1,3\}$, we have $\omega_{P_I} = (1,3,4,6,7,9,10,12,13,15)$. We say that ω_{P_I} has *correct order* if for every $i, j \in I, i < j$, elements of P_i and P_j appear alternatively in $\omega_{P_i \cup P_j}$, starting with an element of P_i. In the example above, ω_{P_I} has correct order.

Consider $\omega \in \mathbb{S}_n$ and suppose that $\omega_{P_i} = (a_1, a_2, \ldots, a_s)$. The elements of the set $P_i = \{a_1, \ldots, a_s\}$ may be viewed as separating subsequences of ω consisting of elements not in P_i. That is, we may write

$$\omega = r_0 a_1 r_1 a_2 \cdots a_s r_s$$

where the r_l's are non-intersecting subsequences of $[n]\backslash P_i$. For each $l, 0 \leq l \leq s$, the subsequence r_l is called the *l-th segment of ω with respect to P_i* and is denoted by $R_l(\omega, P_i)$. Such a segmentation is shown next for the permutation

$$\omega = (c_3, b_4, a_1, b_2, b_3, a_2, a_3, c_1, a_4, b_1, c_2, c_4).$$

Each segment is marked with a bracket:

$$\omega = (\underbrace{c_3, b_4}, a_1, \underbrace{b_2, b_3}, a_2, \underbrace{\ }a_3, \underbrace{c_1}, a_4, \underbrace{b_1, c_2, c_4}).$$

To better visualize the subsequences in question, we may replace each element of P_i by $\star$ and write ω as

$$\omega = (c_3, b_4 \ \star \ b_2, b_3 \ \star \ \star \ c_1 \ \star \ b_1, c_2, c_4).$$

We have, for example, $R_0(\omega, P_i) = (c_3, b_4)$ and $R_2(\omega, P_i) = ()$.

Definition 5.16. Consider $i, j \in [2t+1]$, $P_i = \{a_1, \ldots, a_s\}$, and $\omega \in \mathbb{S}_n$. Suppose, without loss of generality, that $\omega_{P_i} = (a_1, a_2, \ldots, a_s)$. The sequence $\omega_{(j|i)}$ is defined as follows.

- If $i = j$, then $\omega_{(j|i)} = \omega_{(i|i)}$ equals ω_{P_i}.
- If $j > i$, then, for $1 \leq l \leq s$, let $\omega_{(j|i)}(l) = R_l\left(\omega_{P_i \cup P_j}, P_i\right)$ whenever $R_l\left(\omega_{(j|i)}, P_i\right)$ has length one, and let $\omega_{(j|i)}(l) = \epsilon$ otherwise. Here, ϵ is a special notational symbol.
- If $j < i$, then, for $1 \leq l \leq s$, let $\omega_{(j|i)}(l) = R_{l-1}\left(\omega_{P_i \cup P_j}, P_i\right)$ whenever $R_{l-1}\left(\omega_{(j|i)}, P_i\right)$ has length one, and let $\omega_{(j|i)}(l) = \epsilon$ otherwise.

As an example, if $P_i = \{a_1, a_2, a_3, a_4, a_5\}$, $P_j = \{b_1, b_2, b_3, b_4, b_5\}$, $j < i$, and $\omega_{P_i \cup P_j} = (b_1, a_1, a_2, b_2, b_3, a_3, b_4, a_4, b_5, a_5)$, the segments of $\omega_{P_i \cup P_j}$ with respect to P_i are (b_1), $()$, (b_2, b_3), (b_4), and (b_5), in that given order, and we have $\omega_{(j|i)} = (b_1, \epsilon, \epsilon, b_4, b_5)$.

Lemma 5.17. *Consider the interleaved code C of Theorem 5.14 and let $\sigma \in C$. Furthermore, let $\omega \in \mathbb{S}_n$ be such that $\mathsf{d}_U(\sigma, \omega) \leq t$. There exists at least one subset $I \subseteq [2t+1]$ of size at least $t+1$ such that ω_{P_I} has correct order.*

Proof. There are at most t displaced elements, and thus, at most t classes containing a displaced element. Hence, there exist at least $2t+1-t = t+1$ classes without any displaced elements and, consequently, at least $t+1$ σ,ω–pivots. Let I be the set consisting of these pivots. It is clear that ω_{P_I} obtained in this way has correct order which proves the claimed result. □

Lemma 5.18. *For all positive integers s and t and all permutations $\sigma, \omega \in \mathbb{S}_n$, with $n = (2t+1)s$, if i^* is a σ,ω–pivot, then for $j \in [2t+1]$,*

$$\mathsf{d}_H\left(\sigma_{(j|i^*)}, \omega_{(j|i^*)}\right) \leq 2\mathsf{d}_U(\sigma, \omega).$$

Proof. Assume $\mathsf{d}_U(\sigma,\omega) = m$ and let $\psi_m \cdots \psi_2 \psi_1$ be the canonical transform from σ to ω, so that $\omega = \psi_m \cdots \psi_2 \psi_1 \sigma$. We prove the lemma by induction on m. Clearly, if $m = 0$, then

$$\mathsf{d}_H\big(\sigma_{(j|i^*)}, \omega_{(j|i^*)}\big) = 0.$$

Let $\pi = \psi_{m-1} \cdots \psi_2 \psi_1 \sigma$. As the induction hypothesis, assume that

$$\mathsf{d}_H\big(\sigma_{(j|i^*)}, \pi_{(j|i^*)}\big) \leq 2(m-1).$$

By the triangle inequality, it suffices to show that

$$\mathsf{d}_H\big(\pi_{(j|i^*)}, \omega_{(j|i^*)}\big) \leq 2. \tag{5.14}$$

Suppose $\psi_m = \psi(b,\ell)$ so that $\omega = \psi_m \, \pi$. Since i^* is a pivot, we have $b \notin P_{i^*}$. We consider two cases: $b \notin P_j$ and $b \in P_j$. First, suppose $b \notin P_j$. Since $b \notin P_{i^*} \cup P_j$, we have $\pi_{P_{i^*} \cup P_j} = \omega_{P_{i^*} \cup P_j}$ and thus $\mathsf{d}_H\big(\pi_{(j|i^*)}, \omega_{(j|i^*)}\big) = 0$.

On the other hand, suppose $b \in P_j$. Then, b appears in $R_k\left(\pi_{P_{i^*} \cup P_j}, P_{i^*}\right)$ of $\pi_{P_{i^*} \cup P_j}$ and in $R_l\left(\omega_{P_{i^*} \cup P_j}, P_{i^*}\right)$ of $\omega_{P_{i^*} \cup P_j}$, for some k, l. The only segments affected by the translocation ψ_m are $R_k\left(\pi_{P_{i^*} \cup P_j}, P_{i^*}\right)$ and $R_l\left(\omega_{P_{i^*} \cup P_j}, P_{i^*}\right)$, and thus, for $p \in [2t+1] \backslash \{l, k\}$, we have $R_p\left(\pi_{P_{i^*} \cup P_j}, P_{i^*}\right) = R_p\left(\omega_{P_{i^*} \cup P_j}, P_{i^*}\right)$. Hence, for $p \in [2t+1] \backslash \{l, k\}$, we find $\pi_{(j|i^*)}(p) = \omega_{(j|i^*)}(p)$, implying that $\mathsf{d}_H\big(\pi_{(j|i^*)}, \omega_{(j|i^*)}\big) \leq 2$. □

Lemma 5.19. *Consider the interleaved code C of Theorem 5.14. Let $\sigma \in C$ and $\omega \in \mathbb{S}_n$ such that $\mathsf{d}_U(\sigma,\omega) \leq t$. If $I \subseteq [2t+1]$ is of size at least $t+1$ and ω_{P_I} has correct order, then*

1. *for each $i \in I$, $\mathsf{d}_H\left(\sigma_{P_i}, \omega_{P_i}\right) \leq 2t$ and,*
2. *for $i \in I$ and $j \notin I$, $\mathsf{d}_H\left(\sigma_{P_j}, \omega_{(j|i)}\right) \leq 2t$.*

Proof. Since there are at most t classes containing displaced elements and I has size at least $t+1$, there exists a pivot $i^* \in I$. Then, by Lemma 5.18,

$$\mathsf{d}_H\big(\sigma_{(i|i^*)}, \omega_{(i|i^*)}\big) \leq 2t.$$

Since σ is a codeword in C, by construction, we have $\sigma_{(i|i^*)} = \sigma_{P_i}$. Furthermore, since ω_{P_I} has correct order, we have $\omega_{(i|i^*)} = \omega_{P_i}$. Hence,

$$\mathsf{d}_H\left(\sigma_{P_i}, \omega_{P_i}\right) \leq 2t.$$

To prove the second part, we proceed as follows. Assume $\psi_m \cdots \psi_2 \psi_1$, with $m = \mathsf{d}_U(\sigma,\omega)$, is the canonical transform from σ to ω so that $\omega = \psi_m \cdots \psi_2 \psi_1 \sigma$.

We first show that $\psi_m \cdots \psi_2 \psi_1$ may be decomposed into four parts as follows:

$$\omega = \left(\psi^{(j)}_{t^{(j)}} \cdots \psi^{(j)}_1 \left(\psi^{(i)}_{t^{(i)}} \cdots \psi^{(i)}_1 \left(\tau_{t_\tau} \cdots \tau_1 \left(\psi'_{t'} \cdots \psi'_1 \sigma\right)\right)\right)\right),$$

with $t' + t^{(i)} + t^{(j)} = m$ such that

$$\begin{aligned} \psi'_k &= \psi\left(a'_k, \ell'_k\right), \quad a'_k \notin P_i \cup P_j, k \in \left[t'\right], \\ \tau_k &= \left\langle a_k \ b_k \right\rangle, \quad a_k, b_k \in P_i, k \in \left[t_\tau\right], \\ \psi_k^{(i)} &= \psi\left(a_k^{(i)}, \ell_k^{(i)}\right), \quad a_k^{(i)} \in P_i, k \in \left[t^{(i)}\right], \\ \psi_k^{(j)} &= \psi\left(a_k^{(j)}, \ell_k^{(j)}\right), \quad a_k^{(j)} \in P_j, k \in \left[t^{(j)}\right], \end{aligned} \tag{5.15}$$

and such that no $\psi_k^{(i)}$ moves $a_k^{(i)}$ over an element of P_{i^*}.

It can be easily verified that any two translocations $\psi(a, \ell_1)$ and $\psi(b, \ell_2)$ "commute." That is, for any permutation π, we can find translocations $\psi(a, \ell_3)$ and $\psi(b, \ell_4)$ such that $\psi(a, \ell_1)\psi(b, \ell_2)\pi = \psi(b, \ell_4)\psi(a, \ell_3)\pi$. Thus, we have the transform

$$\omega = \left(\psi_{t^{(j)}}^{(j)} \cdots \psi_1^{(j)} \left(\psi''_{t^{(i)}} \cdots \psi''_1 \left(\psi'_{t'} \cdots \psi'_1 \sigma\right)\right)\right)$$

with $t' + t^{(i)} + t^{(j)} = m$ such that

$$\begin{aligned} \psi'_k &= \psi\left(a'_k, \ell'_k\right), \quad a'_k \notin P_i \cup P_j, k \in \left[t'\right], \\ \psi''_k &= \psi(a''_k, \ell''_k), \quad a''_k \in P_i, k \in \left[t^{(i)}\right], \\ \psi_k^{(j)} &= \psi\left(a_k^{(j)}, \ell_k^{(j)}\right), \quad a_k^{(j)} \in P_j, k \in \left[t^{(j)}\right]. \end{aligned}$$

Furthermore, it is easy to see that we may write $\psi''_{t^{(i)}} \cdots \psi''_1$ as $\psi_{t^{(i)}}^{(i)} \cdots \psi_1^{(i)} \tau_{t_\tau} \cdots \tau_1$ with

$$\tau_k = \left\langle a_k \ b_k \right\rangle, \quad a_k, b_k \in P_i, k \in \left[t_\tau\right],$$

such that no $\psi_k^{(i)}$ moves $a_k^{(i)}$ over an element of P_{i^*}. Hence, for any permutation ω one can write a transform of the form (5.15).

Let $\omega' = \tau_{t_\tau} \cdots \tau_1 \psi'_{t'} \cdots \psi'_1 \sigma$, and $\omega^{(i)} = \psi_{t^{(i)}}^{(i)} \cdots \psi_1^{(i)} \omega'$, so that $\omega = \psi_{t^{(j)}}^{(j)} \cdots \psi_1^{(j)} \omega^{(i)}$. By the triangle inequality

$$\begin{aligned} \mathsf{d}_H\left(\sigma_{P_j}, \omega_{(j|i)}\right) \leq\ & \mathsf{d}_H\left(\sigma_{P_j}, \omega'_{(j|i)}\right) \\ & + \mathsf{d}_H\left(\omega'_{(j|i)}, \omega_{(j|i)}^{(i)}\right) \\ & + \mathsf{d}_H\left(\omega_{(j|i)}^{(i)}, \omega_{(j|i)}\right). \end{aligned}$$

It is clear that $\mathsf{d}_H\left(\sigma_{P_j}, \omega'_{(j|i)}\right) = 0$. Next, consider $\omega'_{(j|i)}$ and $\omega_{(j|i)}^{(i)}$. Note that $\omega'_{P_{\{j,i,i^*\}}}$ has correct order. Since no translocation $\psi_k^{(i)}$ moves $a_k^{(i)}$ over an element of P_{i^*}, each $\psi_k^{(i)}$ moves $a_k^{(i)}$ over at most one element of P_j. Thus, each $\psi_k^{(i)}$ can modify at most two segments and we have $\mathsf{d}_H\left(\omega'_{(j|i)}, \omega_{(j|i)}^{(i)}\right) \leq 2t^{(i)}$. Furthermore, each $\psi_k^{(j)}$ modifies at most two segments and thus $\mathsf{d}_H\left(\psi_{(j|i)}^{(i)}, w_{(j|i)}\right) \leq 2t^{(j)}$. Hence,

$$\mathsf{d}_H\left(\sigma_{P_j}, w_{(j|i)}\right) \leq 0 + 2t^{(i)} + 2t^{(j)} \leq 2m.$$

□

Proof. (Theorem 5.14) Suppose the minimum Ulam distance of C is less than $2t+1$. Then, for two distinct codewords $\pi, \sigma \in C$, there exists an $\omega \in \mathbb{S}_n$ such that $\mathsf{d}_U(\pi, \omega) \leq t$ and $\mathsf{d}_U(\sigma, \omega) \leq t$.

Since $\sigma \neq \pi$, there exists $k \in [2t+1]$ such that $\pi_{P_k} \neq \sigma_{P_k}$, which implies that $\mathsf{d}_H(\pi_{P_k}, \sigma_{P_k}) \geq 4t+1$. Since $\mathsf{d}_U(\sigma, \omega) \leq t$, by Lemma 5.17, there exists $I \subseteq [2t+1]$ of size at least $t+1$ such that ω_{P_I} has correct order.

If $k \in I$, by Lemma 5.19, $\mathsf{d}_H(\sigma_{P_k}, \omega_{P_k}) \leq 2t$ and $\mathsf{d}_H(\pi_{P_k}, \omega_{P_k}) \leq 2t$, which together imply $\mathsf{d}_H(\sigma_{P_k}, \pi_{P_k}) \leq 4t$.

On the other hand, if $k \notin I$, by Lemma 5.19, for any $i \in I$, $\mathsf{d}_H\big(\sigma_{P_k}, w_{(k|i)}\big) \leq 2t$ and $\mathsf{d}_H\big(\pi_{P_k}, w_{(k|i)}\big) \leq 2t$, which again imply $\mathsf{d}_H(\sigma_{P_k}, \pi_{P_k}) \leq 4t$.

Hence, by contradiction, the minimum distance of C is at least $2t+1$. □

The rate of translocation correcting codes based on interleaving may be estimated as follows. The cardinality of the interleaved code of length n and minimum distance $d = d(n)$ is at least $\left(A_H\left(\left\lfloor \frac{n}{d} \right\rfloor, 2d-1\right)\right)^d$ for odd d, and $\left(A_H\left(\left\lfloor \frac{n}{d+1} \right\rfloor, 2d+1\right)\right)^{d+1}$ for even d. The construction is applicable only if $d(n) \leq \sqrt{n/2} - 1$, in which case the asymptotic rate of the interleaved code equals

$$\begin{aligned}
\lim \frac{\ln |C|}{\ln n!} &= \lim \frac{d(n) \ln A_H\left(\frac{n}{d(n)}, 2d(n)\right)}{\ln n!} \\
&= \lim \frac{\ln A_H\left(\frac{n}{d(n)}, 2d(n)\right)}{\ln (n/d(n))!} \frac{d(n) \ln (n/d(n))!}{\ln n!} \\
&= \left(1 - 2 \lim \frac{d^2(n)}{n}\right) \lim \frac{n \ln n - n \ln d(n) + O(n)}{n \ln n + O(n)},
\end{aligned}$$

where we have used Theorem 5.8 to obtain the last equality. For example, if $d(n) = n^\beta$, $\beta < 1/2$, then

$$1 - 2 \lim \frac{d^2(n)}{n} = 1$$

and one obtains a translocation error-correcting code of rate

$$\lim \frac{\ln |C|}{\ln n!} = \lim \frac{n \ln n - \beta n \ln n + O(n)}{n \ln n + O(n)} = 1 - \beta.$$

In the next section we describe a modification of the interleaving procedure, which, when applied recursively, improves upon the code rate $1 - \beta$.

5.5.2 Interleaving Codes in the Hamming and Ulam Metrics

The interleaving approach described in the previous subsection may be extended in a straightforward manner: Rather than interleaving permutation codes with good Hamming distance, as in §5.5.1, one may construct a code in the Ulam metric by interleaving a code with good Ulam distance and a code with good

Hamming distance. Furthermore, this approach may be implemented in a recursive manner. In what follows, we explain one such approach and show how it leads to improved code rates as compared to simple interleaving.

We find the following results useful for our recursive construction method.

Lemma 5.20. *Let $\sigma, \pi \in \mathbb{S}_n$ be two permutations, such that $\mathsf{d}_U(\sigma, \pi) = 1$. Then, there exist at most three locations i, $i \in [n-1]$, such that for some $j = j(i) \in [n-1]$:*

- $\sigma(i) = \pi(j)$;
- $\sigma(i+1) \neq \pi(j+1)$.

Proof. Suppose $\pi = \sigma\phi(i_1, i_2)$. The proof follows from the simple fact that when applying a translocation $\phi(i_1, i_2)$ to σ, the locations j described above are among

$$\begin{cases} i_1, i_2 - 1, \text{ and } i_2, & \text{if } i_1 > i_2, \\ i_1 - 1, i_2 - 1, \text{ and } i_2, & \text{if } i_1 < i_2. \end{cases}$$

□

Corollary 5.21. *Let $\sigma, \pi \in \mathbb{S}_n$ be two permutations, and assume that there exist $a \geq 0$ different locations i, $i \in [n-1]$, such that $\sigma(i) = \pi(j)$, but $\sigma(i+1) \neq \pi(j+1)$ for some $j \in [n-1]$. Then, $\mathsf{d}_U(\sigma, \pi) \geq \lceil a/3 \rceil$.*

For an integer $p \geq 1$, let $\mu = (1, 2, \ldots, p)$ and let $\sigma_1, \sigma_2 \in \mathbb{S}(\{p+1, \ldots, 2p-1\})$. Note that

$$\begin{aligned} \mu \circ \sigma_1 &= (1, \sigma_1(1), 2, \sigma_1(2), \ldots, p-1, \sigma_1(p-1), p) \\ \mu \circ \sigma_2 &= (1, \sigma_2(1), 2, \sigma_2(2), \ldots, p-1, \sigma_2(p-1), p). \end{aligned} \tag{5.16}$$

Theorem 5.22. *For $\mu, \sigma_1,$ and σ_2 described above, if $\mathsf{d}_H(\sigma_1, \sigma_2) \geq d$, then*

$$\mathsf{d}_U(\mu \circ \sigma_1, \mu \circ \sigma_2) \geq \lceil 2d/3 \rceil .$$

Proof. Let $\pi_1 = \mu \circ \sigma_1$ and $\pi_2 = \mu \circ \sigma_2$. Below, we show that the number of indices ℓ in π_1, with respect to π_2, that satisfy the conditions described in Lemma 5.20 is at least $2d$. Then, the claim of the theorem follows when we apply Corollary 5.21 with $a = 2d$.

Assume that $\sigma_1(\ell) \neq \sigma_2(\ell)$ for some $\ell \in [p-1]$. For each such ℓ, the two indices $2\ell - 1$ and 2ℓ can both serve as index i in Lemma 5.20:

1. We have $\pi_1(2\ell - 1) = \pi_2(2\ell - 1) = \ell$, yet

$$\sigma_1(\ell) = \pi_1(2\ell) \neq \pi_2(2\ell) = \sigma_2(\ell).$$

2. Let $j \in [p]$ be such that $\pi_1(2\ell) = \pi_2(2j)$. It is easy to see that $j \neq \ell$. Then,

$$\ell + 1 = \pi_1(2\ell + 1) \neq \pi_2(2j + 1) = j + 1 .$$

□

Let $\mu \circ C = \{\mu \circ \sigma : \sigma \in C\}$. The following corollary follows from Theorem 5.22.

Corollary 5.23. *For integers n and p with $n = 2p-1$, let $\mu = (1, 2, \ldots, p)$ and suppose $C \subseteq \mathbb{S}(\{p+1, \ldots, n\})$ is a code with minimum Hamming distance at least $\frac{3d}{2}$. Then $\mu \circ C$ is a code in $\mathbb{S}_n$ with minimum Ulam distance at least d and with size $|C|$.*

Hence, for odd n, we can construct a translocation code with length n, minimum distance at least d, and size $A_H\left(\frac{(n-1)}{2}, \left\lceil \frac{3d}{2} \right\rceil\right)$. This can be easily generalized for all n to get codes of size

$$A_H\left(\left\lceil \frac{n}{2} \right\rceil - 1, \left\lceil \frac{3d}{2} \right\rceil\right).$$

By assuming that the permutation code in the Hamming metric is capacity achieving, the asymptotic rate of the constructed code becomes

$$\begin{aligned}
&\lim \frac{\ln A_H\left(\left\lceil \frac{n}{2} \right\rceil - 1, \left\lceil \frac{3d(n)}{2} \right\rceil\right)}{\ln n!} \\
&= \lim \frac{\ln A_H\left(\left\lceil \frac{n}{2} \right\rceil - 1, \left\lceil \frac{3d(n)}{2} \right\rceil\right)}{\ln \left\lceil \frac{n}{2} \right\rceil!} \cdot \frac{\ln \left\lceil \frac{n}{2} \right\rceil!}{\ln n!} \\
&= \frac{1}{2} - \frac{3}{2} \lim \frac{d(n)}{n} = \frac{1}{2} - \frac{3}{2}\delta,
\end{aligned}$$

where $\delta = \lim \frac{d(n)}{n}$. Therefore, this code construction incurs a rate loss of $1/2(1+\delta)$ when compared to the capacity bound, which in this case equals $1-\delta$.

The final result that we prove in order to describe a recursive interleaving procedure is related to the longest common subsequence of two sequences and the minimum Ulam distance of interleaved sequences.

Lemma 5.24. *For $\sigma, \pi \in \mathbb{S}_n$ and $P \subseteq [n]$, we have*

$$\mathsf{d}_U(\sigma, \pi) \geq \mathsf{d}_U(\sigma_P, \pi_P) + \mathsf{d}_U(\sigma_Q, \pi_Q),$$

where $Q = [n] \backslash P$.

Proof. Without loss of generality, assume that σ is the identity permutation. It is clear that $l(\pi) \leq l(\pi_P) + l(\pi_Q)$. Hence,

$$\begin{aligned}
\mathsf{d}_U(\sigma, \pi) &= n - l(\pi) \\
&\geq n - l(\pi_P) - l(\pi_Q) \\
&= |P| - l(\pi_P) + |Q| - l(\pi_Q) \\
&= \mathsf{d}_U(\sigma_P, \pi_P) + \mathsf{d}_U(\sigma_Q, \pi_Q).
\end{aligned}$$

□

Lemma 5.25. *For sets P and Q of sizes p and $p-1$, respectively, let $C_1' \subseteq \mathbb{S}(P)$ be a code with minimum Ulam distance d and let $C_1 \subset \mathbb{S}(Q)$ be a code with minimum Hamming distance $3d/2$. The code $C_1' \circ C_1 = \{\sigma \circ \pi : \sigma \in C_1', \pi \in C_1\}$ has minimum Ulam distance d.*

Proof. For $\sigma_1, \sigma_2 \in C_1'$ and $\pi_1, \pi_2 \in C_1$ with $(\sigma_1, \pi_1) \neq (\sigma_2, \pi_2)$, we show that $\mathsf{d}_U(\sigma_1 \circ \pi_1, \sigma_2 \circ \pi_2) \geq d$.

The case $\sigma_1 = \sigma_2$ follows from a simple use of Theorem 5.22.

Assume next that $\sigma_1 \neq \sigma_2$. Then by Lemma 5.24, $\mathsf{d}_U(\sigma_1 \circ \pi_1, \sigma_2 \circ \pi_2) \geq \mathsf{d}_U(\sigma_1, \sigma_2) \geq d$ and this completes the proof. □

We now turn out attention to a recursive code construction based on the findings outlined above.

Let $\alpha = \frac{3}{2}$. For a given n, set $P = \left\{1, \ldots, \left\lceil \frac{n}{2} \right\rceil\right\}$ and set $Q = \left\{\left\lceil \frac{n}{2} \right\rceil + 1, \ldots, 2\left\lceil \frac{n}{2} \right\rceil - 1\right\}$. Suppose $C_1' \subseteq \mathbb{S}(P)$ is a code with minimum Ulam distance d and $C_1 \subseteq \mathbb{S}(Q)$ a code with minimum Hamming distance $\alpha\, d$. Assuming that permutation codes with this given minimum Hamming distance exist, we only need to provide a construction for C_1'. An obvious choice for C_1' is a code with only one codeword. Then, $C = C_1' \circ C_1$ is a code with minimum Ulam distance d and cardinality

$$A_H\left(\left\lceil \frac{n}{2} \right\rceil - 1, \alpha\delta\right).$$

The gap to capacity may be significantly reduced by observing that C_1' does not have to be a code of cardinality one, and that C_1' may be constructed from shorter codes.

To this end, let $C_1' = C_2' \circ C_2$ where C_2' is a code of length $\left\lceil \frac{n}{4} \right\rceil$ with minimum Ulam distance d, while C_2 is a code of length $\left\lceil \frac{n}{4} \right\rceil - 1$ and minimum Hamming distance αd.

By repeating the same procedure k times we obtain a code of the form

$$(((C_k' \circ C_k) \circ C_{k-1}) \circ \cdots) \circ C_1, \tag{5.17}$$

where each C_i, $i \leq k$, is a code with minimum Hamming distance αd and length $\left\lceil \frac{n}{2^i} \right\rceil - 1$, and C_k' is a code with minimum Ulam distance d and length $\left\lceil \frac{n}{2^i} \right\rceil$. Since each C_i is a permutation code in the Hamming metric with minimum distance αd, we must have $\left\lceil \frac{n}{2^i} \right\rceil - 1 \geq \alpha d$. To ensure that this condition is satisfied, in (5.17), we let k be the largest value of i satisfying $\frac{n}{2^i} - 1 \geq \alpha d$. It is easy to see that $k = \left\lfloor \log \frac{n}{\alpha d + 1} \right\rfloor$. Furthermore, we choose C_k' to consist of a single codeword.

The asymptotic rate of the recursively constructed codes equals

$$\begin{aligned}
&\lim \frac{1}{\ln n!} \sum_{i=1}^{k} \ln A_H \left(\left\lceil \frac{n}{2^i} \right\rceil - 1, \alpha d(n) \right) \\
&= \lim \sum_{i=1}^{k} \frac{\ln A_H \left(\left\lceil \frac{n}{2^i} \right\rceil - 1, \alpha d(n) \right)}{\ln \left(\left\lceil \frac{n}{2^i} \right\rceil - 1 \right)!} \frac{\ln \left(\left\lceil \frac{n}{2^i} \right\rceil - 1 \right)!}{\ln n!} \\
&= \lim \sum_{i=1}^{k} \left(1 - \frac{\alpha d(n) 2^i}{n} \right) 2^{-i} \\
&= \lim \left(1 - 2^{-k} - \frac{\alpha d(n) k}{n} \right) \\
&= 1 - 2^{-\lfloor \log \frac{1}{\alpha\delta} \rfloor} - \alpha\delta \left\lfloor \log \frac{1}{\alpha\delta} \right\rfloor,
\end{aligned}$$

where the last step follows from $\lim k = \left\lfloor \log \frac{1}{\alpha\delta} \right\rfloor$. Note that this rate is roughly equal to $1 - \alpha\delta \left(1 + \log \frac{1}{\alpha\delta} \right)$.

5.5.3 Permutation Codes in the Hamming Metric

In the previous subsection, we demonstrated a number of constructions for translocation error-correcting codes based on permutation codes in the Hamming metric and codes over distinct symbols. There exists a number of constructions for sets of permutations with good Hamming distance, and codes with codewords containing distinct symbols. For example, in [94–97] constructions of permutations in $\mathbb{S}_n$ using classical binary codes were presented, while other constructions rely on direct combinatorial arguments [98,99]. An example of code construction for codewords over distinct symbols was presented in [93]. There, specialized subcodes of Reed-Solomon codes were identified such that their codewords consist of distinct symbols.

In the former case, if C is a binary $[n, \alpha n, \beta n]$ code, the construction applied to C yields a subset of $\mathbb{S}_n$ of cardinality $2^{\alpha n}$, with minimum Hamming distance βn. This construction and constructions related to it may be used for permutation code design, resulting in sets of permutations in $\mathbb{S}_n$ of cardinality $\exp\{\Theta(n)\}$ and minimum Hamming distance $\Theta(n)$. These permutations may consequently be used to construct permutation codes in $\mathbb{S}_{2n}$ with $\exp\{\Theta(n)\}$ codewords and minimum Ulam distance $\Theta(n)$.

Next, we describe a simple method for constructing sets of vectors of length $m > 0$ over $[n]$ such that all entries of the vector are different, and such that the minimum Hamming distance between the vectors is large. In other words, we propose a novel construction for *partial permutation codes* under the Hamming metric, suitable for use in the recursive code construction described in the previous subsection.

The idea behind the proof is based on mapping suitably modified binary codewords in the Hamming space into partial permutations. For this purpose,

let C be a binary $[N, K, D]$ code, and for simplicity of exposition, assume that n is a power of two. Let $\boldsymbol{c} \in C$. We construct a vector $\boldsymbol{x} = \chi(\boldsymbol{c}) \in ([n])^m$, where χ is a mapping described below, as follows:

1. Divide $\boldsymbol{c}$ into m binary blocks $\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_m$ of lengths $\log_2 n - \log_2 m$ each. Again, for simplicity, we assume that m is a power of two.

2. For each block $\boldsymbol{c}_i$, $i \in [m]$, construct a vector $\boldsymbol{x}_i$ of length $\log_2 n$ according to the rule: the first $\log_2 n - \log_2 m$ bits in $\boldsymbol{x}_i$ equal $\boldsymbol{c}_i$, while the last $\log_2 m$ bits in $\boldsymbol{x}_i$ represent the binary encoding of the index i. Note that the integer values represented by the binary vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$ are all different.

3. Form an integer valued vector $\boldsymbol{x} = \chi(\boldsymbol{c})$ of length m over $[n]$, such that its i-th entry has the binary encoding specified by $\boldsymbol{x}_i$. Observe that all the integer entries of such a vector are different.

Now, take two vectors $\boldsymbol{a}, \boldsymbol{c} \in C$, such that their Hamming distance satisfies $\mathsf{d}_H(\boldsymbol{a}, \boldsymbol{c}) \geq D$. Let $\boldsymbol{x} = \chi(\boldsymbol{a})$ and $\boldsymbol{y} = \chi(\boldsymbol{c})$ be the corresponding vectors of length m over $[n]$ constructed as above. Then, there exist at least $\frac{D}{\log_2 n - \log_2 m}$ blocks of length $\log_2 n$ that are pairwise different. Therefore, the corresponding $\frac{D}{\log_2 n - \log_2 m}$ entries in $\boldsymbol{x}$ and $\boldsymbol{y}$ are pairwise different as well.

Consider the set of vectors

$$\mathcal{S}' = \{\chi(\boldsymbol{c}) \,:\, \boldsymbol{c} \in C\} \ .$$

It is straightforward to see that the set $\mathcal{S}'$ has the following properties:

1. For any $\boldsymbol{x} \in \mathcal{S}'$, all entries in $\boldsymbol{x}$ are different.

2. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}'$, $\boldsymbol{x} \neq \boldsymbol{y}$, the Hamming distance satisfies $\mathsf{d}_H(\boldsymbol{x}, \boldsymbol{y}) \geq \frac{D}{\log_2 n - \log_2 m}$.

The set $\mathcal{S}'$ can be used similarly as the set $\mathbb{S}_n$ in the basic construction to obtain codes over $\mathbb{S}_{n+m}$ with Ulam distance

$$O\left(\frac{D}{\log_2 n - \log_2 m}\right) .$$

Note that in this case, only m numbers in the range $\{n+1, n+2, \cdots, n+m\}$ are inserted between the numbers in $[n]$, while the Hamming distance of the vectors is preserved.

Lemma 5.26. *The parameters N, n and m are connected by the following equation:*

$$N = m \log_2 n - m \log_2 m = m \log_2 \frac{n}{m} \ .$$

From this lemma, if we take $m = \frac{1}{2}n$, then $N = \frac{1}{2}n$. By taking a code C with parameters $[\frac{1}{2}n, \alpha n, \beta n]$, where $\alpha > 0$ and $\beta > 0$ are constants, we obtain a set

$\mathcal{S}'$ of size $2^{\alpha n}$ and Hamming distance $\Theta(n)$. The corresponding code is able to correct $\Theta(n)$ translocation errors, and it has $2^{\alpha n}$ codewords.

5.5.4 Decoding of Interleaved Codes

An efficient decoder implementation for the general family of interleaved codes is currently not known. For the case of recursive codes, decoding may be accomplished with low complexity provided that the Hamming distance of the component permutation codes is increased from $\frac{3d}{2}$ to $2d$.

For simplicity of exposition, we assume $n = 2n' + 1$. The case of even n may be handled in the same manner, provided that one fixes the last symbol of all codewords.

Let $\sigma = (1, \hat{\sigma}(1), 2, \hat{\sigma}(2), \ldots, \hat{\sigma}(n'), n'+1) \in C$ be the stored codeword and let $\pi \in \mathbb{S}_n$ be the retrieved word.

For $i \in [n']$, denote by s_i^{π} the substring of π that starts with element i and ends with element $i+1$. If $i+1$ appears before i in π, then s_i^{π} is considered empty. For $i \in [n']$, let $\hat{\pi}(i) = u$ if s_i^{π} contains a unique element of $[n] \backslash [n']$, say u. Otherwise, let $\hat{\pi}(i) = \epsilon$.

Lemma 5.27. *The permutation $\hat{\pi}$ differs from $\hat{\sigma}$ in at most $2\mathsf{d}_U(\sigma, \pi)$ positions.*

Proof. Let $t = \mathsf{d}_U(\sigma, \pi)$. There exists a sequence $\phi_1, \phi_2, \ldots, \phi_t$ of translocations such that $\pi = \sigma\phi_1\phi_2\cdots\phi_t$. For $i \in \{0, \ldots, t\}$, let $\pi_i = \sigma\phi_1\phi_2\cdots\phi_i$ and let L_i be given as

$$L_i = \{j | \exists k \leq i : \hat{\pi}_k(j) \neq \hat{\sigma}_k(j)\}.$$

The set L_i may be viewed as the set of elements displaced by one of the translocations $\phi_1, \phi_2, \ldots, \phi_i$. Note that, for each i, $L_i \subseteq L_{i+1}$.

To prove the lemma, it suffices to show that $|L_t| \leq 2t$, since $\{j | \hat{\pi}(j) \neq \hat{\sigma}(j)\} \subseteq L_t$.

Let $L_0 = \varnothing$. We show that $|L_i| \leq |L_{i-1}| + 2$ for $i \in [t]$.

The translocation ϕ_i either moves an element of $[n']$ or an element of $[n] \backslash [n']$. First, suppose that it moves an element j of $[n']$. Then ϕ_i can affect only the substrings $s_{j-1}^{\pi_{i-1}}$ and $s_j^{\pi_{i-1}}$ of π_{i-1}. In the second case, assume that ϕ_i moves an element of $[n] \backslash [n']$. It can then be easily verified that at most two substrings of π_{i-1} may be affected by the given translocation. Hence, $|L_i| \leq |L_{i-1}| + 2$. □

Assume now that $C \subseteq \mathbb{S}_n$ is an interleaved code of the form

$$C = C_1' \circ C_1,$$

where $C_1' = \{(1, 2, \ldots, n'+1)\}$, and where C_1 is a permutation code over the set $P = \{n'+2, \ldots, 2n'+1\}$ with minimum Hamming distance $4t+1$.

Let $\sigma \in C$ be the stored code word and $\pi \in \mathbb{S}_n$ be the retrieved word. Assume that $\mathsf{d}_U(\sigma, \pi) \leq t$. The first step of the decoding algorithm is to extract $\hat{\pi}$ from

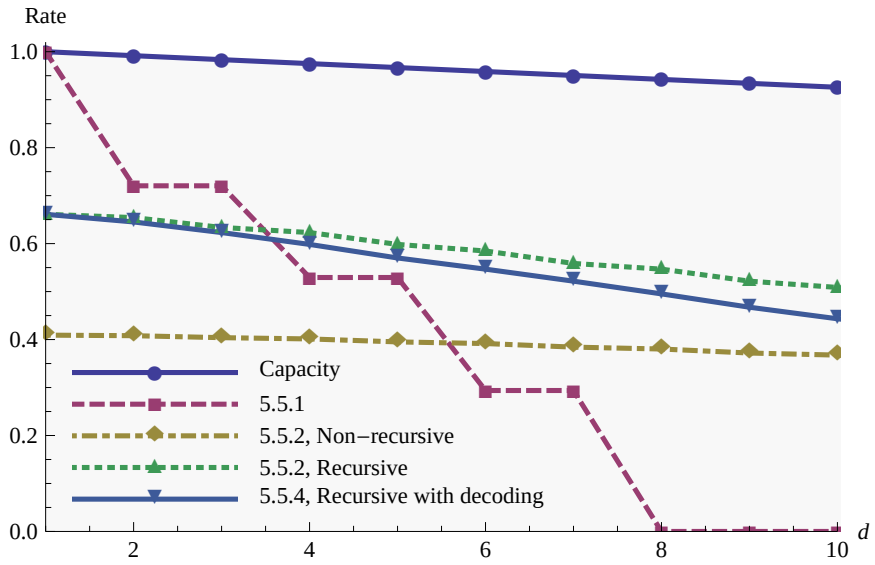


Figure 5.4: Rate vs. distance for several code constructions with $n = 150$. The numbers in the legend refer to the section where the corresponding code is described. It is assumed that $A_H(n,d) = \frac{n!}{(d-1)!}$.

the permutation π. By Lemma 5.27, we have $\mathsf{d}_H(\hat{\sigma},\hat{\pi}) \leq 2t$. Since C_1 has minimum Hamming distance $4t+1$, $\hat{\sigma}$ can be uniquely recovered from $\hat{\pi}$. Thus, decoding is accomplished through Hamming distance decoding of permutation codes, for which a number of interesting algorithms are known in literature [88, 89, 100]. Note that this simple decoding method may be used on a recursive construction of depth larger than one, following the same rules as described before.

The asymptotic rate of the code C is $\frac{1}{2} - 2\delta$, where $\delta = \lim \frac{d}{n} = \lim \frac{2t+1}{n}$. For the recursive construction described in (5.17), the asymptotic rate of the efficiently decodable codes outlined above equals $1 - 2^{-\lfloor \log \frac{1}{\alpha\delta} \rfloor} - \alpha\delta \left\lfloor \log \frac{1}{\alpha\delta} \right\rfloor$, with $\alpha = 2$.

Remark: Permutation codes in $\mathbb{S}_n$, correcting adjacent transposition errors, were thoroughly studied in [36]. We note that these codes can also be used to correct translocation errors. Indeed, every translocation can be viewed as a sequence of at most $n-1$ adjacent transpositions. Therefore, any code in $\mathbb{S}_n$ that corrects $f(n)$ adjacent transpositions (for some function $f(n)$) can also correct $O(f(n)/n)$ translocations.

It was shown in Theorem 3.1 of [36] that the upper bound on the rate of the code correcting $O(n^2)$ adjacent transpositions is zero. Such a code can also be used to correct $O(n)$ translocation errors. In comparison, the interleaved constructions described above can also correct $O(n)$ translocation errors, yet their rate is strictly larger than zero.

The non-asymptotic and asymptotic rates of the discussed code families are compared in Figures 5.4 and 5.5.

Note that the gap from capacity of the constructions presented in this chapter

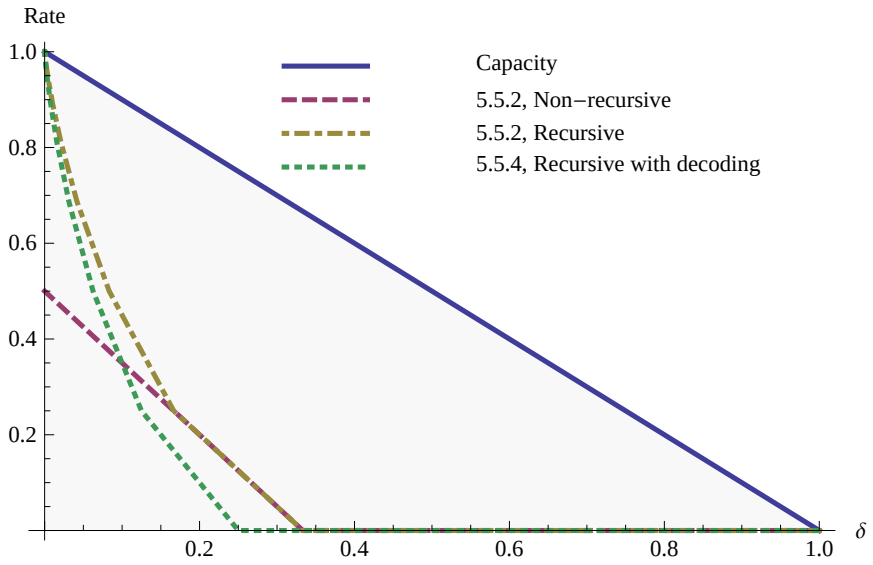


Figure 5.5: Asymptotic rate vs. distance for several code constructions. The horizontal axis is $\delta = \lim \frac{d(n)}{n}$.

is still fairly large, despite the fact that the codes are asymptotically good. This result may be attributed to the fact that the interleaving construction restricts the locations of subsets of elements in a severe manner. It is worthwhile to study Alternative interleaving methods that may lead to codes with higher rates.

In what follows, we describe a method of Beame et al. [85] that provides translocation codes with minimum distance proportional to $n - o(n)$. This covers the zero-capacity domain of our analysis.

5.5.5 Zero-rate Codes

We present two constructions based on the longest common subsequence analysis. The first construction is based on Hadamard matrices was and given in [85], while the second construction is probabilistic.

Assume that a Hadamard matrix of order k exists. To explain the construction, we consider permutations over the set $\{0, 1, \ldots, n-1\}$. Furthermore, the positions in each permutation are also numbered from 0 to $n-1$.

Let $s = \lceil n^{1/(k-1)} \rceil$. For $a \in \{0, 1, \ldots, n-1\}$, we denote the representation of a in base s by $\overline{a_1 a_2 \cdots a_{k-1}}$, where a_1 is the most significant digit.

Let H be a Hadamard matrix of order k with rows and columns indexed by elements in the set $\{0, 1, \ldots, k-1\}$. Without loss of generality, assume the first row and column of H are all-ones vectors. The set $\{\pi_i\}_{i=1}^{k}$ of permutations is constructed by defining the mth element of π_i, for $m = 0, 1, \ldots, s^{k-1} - 1$, as follows. Let $m = \overline{m_1 \cdots m_{k-1}}$, and let the mth element of π_i equal

$$\pi_i(m) = \overline{a_1 a_2 \cdots a_{k-1}},$$

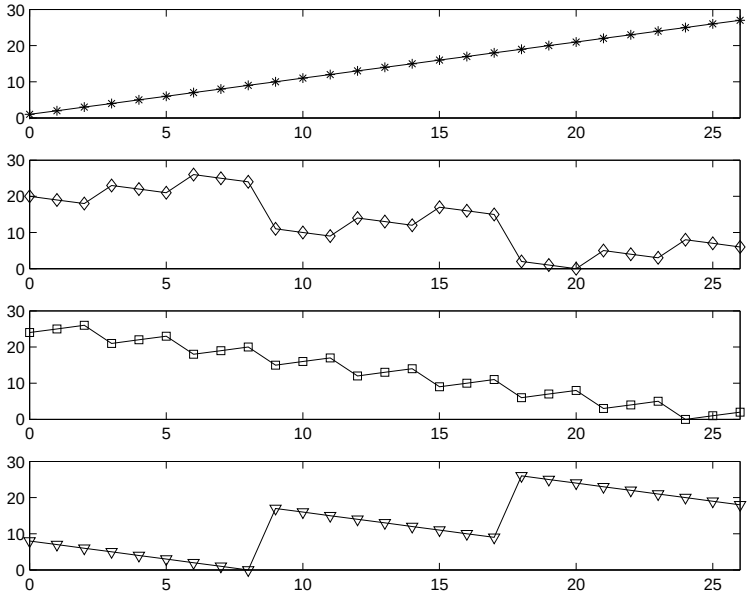


Figure 5.6: Permutation codewords based on Hadamard matrices [85].

where, for $j \in \{0, 1, \ldots, k-1\}$,

$$a_j = \begin{cases} m_j, & \text{if } H_{ij} = 1, \\ s - 1 - m_j, & \text{if } H_{ij} = -1. \end{cases}$$

The length of the longest common subsequence of any two permutations of $\{\pi_i\}$ is at most $s^{k/2-1}$. The permutations obtained in this way have length s^{k-1}. Consequently, the minimum distance of the code is at least $s^{k-1} - s^{k/2-1}$. Note that if $s^{k-1} > n$, we can arbitrarily delete elements from each permutation to obtain a set of permutations each of length n.

As an example, consider $n = 27$ and $k = 4$. We have

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

and $s = 3$. Four codewords of the code based on this Hadamard matrix are plotted in Figure 5.6.

Another construction based on [85] holds for $3 \le k \le \sqrt{n}$, leading to k permutations with minimum Ulam distance at least $n - 32\,(kn)^{1/3}$. The number of codewords obtained from this construction is exponentially smaller than what may be obtained via random methods, as we demonstrate next.

Let U_n denote the Ulam distance between a uniformly and at random chosen permutation of length n and the identity, $e = (1, 2, \ldots, n)$. From a result shown by Kim [101] (see also [102–104]), for $0 < \theta \leq n^{1/3}/20$, one has

$$\begin{aligned} &P\left(U_n \leq n - 2\sqrt{n} - \theta n^{1/6}\right) \\ &\leq \exp\left(-\theta^{3/2}\left(\frac{4}{3} - \frac{\theta}{27n^{1/3}} - \frac{5\log n}{\theta^{1/2}n^{1/3}}\right)\right). \end{aligned}$$

By letting $\theta = an^{1/3}$ with $a \leq 1/20$, for sufficiently large n, we find

$$P\left(U_n \leq n - (2+a)\sqrt{n}\right) \leq \exp\left(-a^{3/2}\sqrt{n}\right).$$

Suppose a code C is constructed by randomly choosing $M = e^{\alpha_n}$ permutations in $\mathbb{S}_n$, with replacement. By left-invariance, the bound above also holds for the Ulam distance between two given codewords of C. Using the union bound and the fact that there are less than M^2 pairs of codewords, the probability that there exist two permutations with distance $\leq n - (2+a)\sqrt{n}$ is bounded from above by

$$M^2 P\left(U_n \leq n - (2+a)\sqrt{n}\right) \leq \exp\left(-a^{3/2}\sqrt{n} + 2\alpha_n\right).$$

To ensure that the minimum distance of the code is at least $n - (2+a)\sqrt{n}$ with high probability, we must choose α_n such that $a^{3/2}\sqrt{n} > 2\alpha_n$. Hence, we let $\alpha_n = \frac{1}{2}\sqrt{a^3 n} - \epsilon$, for some $\epsilon > 0$. For this choice, with high probability, the random code C of size $\Theta\left(e^{\sqrt{a^3 n}/2}\right)$ has minimum distance at least $n - (2+a)\sqrt{n}$. In particular, for $a = 1/20$, there exists a code of size $\Theta\left(e^{\sqrt{n/5}/80}\right)$ with minimum distance at least $n - 2.05\sqrt{n}$.

As already pointed out, the size of a randomly constructed code obtained this way is exponential in $\sqrt{n}$, while the size of the code from the explicit construction in [85],

$$\left(\frac{2+a}{32}\right)^3 \sqrt{n},$$

is only linear in $\sqrt{n}$.

5.6 Summary

We introduced the notion of translocation errors in rank modulation systems. Translocation errors may be viewed as generalization of adjacent swap errors frequently encountered in flash memories. We demonstrated that the metric used to capture the effects of translocation errors is the Ulam distance between two permutations, a linear function of the longest common subsequence of the permutations. We also derived asymptotically tight upper and lower bounds on the code capacity. Furthermore, we presented a number of constructions for

translocation error-correcting codes based on interleaving permutation codes in the Hamming metric and deletion-correcting codes in the Levenshtein metric. Finally, we exhibited a low-complexity decoding method for a class of relaxed interleaved codes of non-zero asymptotic rate.

Chapter 6

Conclusion and Future Work

Distance functions on rankings (permutations) play an important role in mathematics, the social sciences, bioinformatics, and coding theory. Our work established a new framework in which to study very general forms of such distance measures, and we also described a number of modern applications of the metrics in the context of rank aggregation and rank coding for flash memories.

Using an axiomatic approach, we introduced two novel classes of distances on rankings, and proposed algorithms for computing or approximating these distances. In addition, we described efficient aggregation algorithms for these distances. For flash memories, we introduced codes in a distance metric suitable for capturing deletion and insertion errors, and consequently, arbitrary charge drops. The proposed error-correcting schemes include capacity achieving codes that can correct a constant number of errors, asymptotically good codes, as well as simple decoding algorithms.

There remain many problems to be addressed in the future. While we have found algorithms for computing the distances for some of the most important cases, general algorithms are not yet known. At the moment, we are in the process of extending our results for metric trees [105]. Additionally, since rank aggregation with many distances is an NP-hard problem, further research is required for obtaining good approximation algorithms. Approaches based on linear programming relaxations appear to be particularly promising, and our recently submitted work is available at [106]. We are also considering iterative vote aggregation models in which voters may change their votes after observing the votes of their peers. See [107]. Since our focus was on theoretical analysis of the problem, it remains to demonstrate the use of our methods in practice. Currently, we are in the process of applying weighted distances to ordinal genomic data fusion [108].

The codes that we proposed for flash memories are based on interleaving shorter codes. Interleaving limits the positions that subsets of elements can occupy in the code. Further research is needed to find out if this limitation can be relaxed while maintaining good minimum distance. Other interesting topics related to rank codes are the study of limited-magnitude errors as well as adapting rank codes for distributed storage systems. These are some of the subjects that we intend to study in the future.

Bibliography

[1] National Basketball Association, "NBA Standings." [Online]. Available: http://www.nba.com/standings/

[2] "puppywar.com." [Online]. Available: http://puppywar.com/

[3] "kittenwar.com." [Online]. Available: http://kittenwar.com/

[4] U.S. News & World Report, "Best undergraduate engineering programs rankings." [Online]. Available: http://colleges.usnews.rankingsandreviews.com/best-colleges/rankings/engineering-doctorate

[5] "Topspeed's 10 best mid-level sports cars." [Online]. Available: http://www.topspeed.com/cars/car-news/topspeed-s-10-best-mid-level-sports-cars-ar94691.html

[6] CNN Money, "100 best companies to work for." [Online]. Available: http://money.cnn.com/magazines/fortune/best-companies/?iid=bc_fl_header

[7] "listverse.com." [Online]. Available: http://listverse.com/

[8] Z. Udko, "The top 10 top 10 lists of 2012." [Online]. Available: http://www.huffingtonpost.com/zach-udko/top-10-lists-2012_b_2367506.html

[9] M. Kendall, *Rank correlation methods*. London: Griffin, 1948.

[10] J. Marden, *Analyzing and Modeling Rank Data*, ser. Monographs on Statistics and Applied Probability. Chapman & Hall, 1995. [Online]. Available: http://books.google.com/books?id=Dp24H21QHNoC

[11] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, 1952. [Online]. Available: http://www.jstor.org/stable/2280779

[12] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation revisited," Manuscript, Available: www.eecs.harvard.edu/~michaelm/CS222/rank2.pdf, 2001.

[13] M. Regenwetter, B. Grofman, A. Marley, and I. Tsetlin, *Behavioral social choice: probabilistic models, statistical inference, and applications*. Cambridge University Press, 2006.

[14] H. Young, "Social choice scoring functions," *SIAM J. Appl. Math.*, vol. 28, no. 4, pp. 824–838, 06 1975.

[15] J. G. Kemeny, "Mathematics without numbers," *Daedalus*, vol. 88, no. 4, pp. 577–591, 1959.

[16] J. G. Kemeny and J. Snell, *Mathematical models in the social sciences*. Boston: Ginn, 1962.

[17] E. Ephrati and J. Rosenschein, "Multi-agent planning as a dynamic search for social consensus," in *Proc. 13 Int. Joint Conf. on Artificial Intelligence (IJCAI)*, vol. 1, San Mateo, CA, 1993, pp. 423–9.

[18] D. M. Pennock, E. Horvitz, C. L. Giles et al., "Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering," in *Proc. Nat. Conf. Artificial Intelligence*, Austin, Texas, 2000, pp. 729–734.

[19] A. J. H. Vinck, "Coded modulation for power line communications," *Arxiv preprint arXiv:1104.1528*, 2011.

[20] A. Jiang, M. Schwartz, and J. Bruck, "Error-correcting codes for rank modulation," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, Canada, July 2008, pp. 1736–1740.

[21] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. Information Theory*, vol. 55, no. 6, pp. 2659–2673, June 2009.

[22] H. Young and A. Levenglick, "A consistent extension of Condorcet's election principle," *SIAM J. Appl. Math.*, vol. 35, no. 2, pp. 285–300, 1978. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/0135023

[23] P. Diaconis and R. Graham, "Spearman's footrule as a measure of disarray," *J. Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 2, pp. 262–268, 1977.

[24] I. Salama and D. Quade, "A nonparametric comparison of two multiple regressions by means of a weighted measure of correlation," *Communications in Statistics - Theory and Methods*, vol. 11, no. 11, pp. 1185–1195, 1982. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/03610928208828304

[25] G. S. Shieh, "A weighted Kendall's tau statistic," *Statistics and Probability Letters*, vol. 39, no. 1, pp. 17 – 24, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167715298000066

[26] E. Yilmaz, J. A. Aslam, and S. Robertson, "A new rank correlation coefficient for information retrieval," in *Proc. 31st Annu. Int. SIGIR Conf. Research and Development in Information Retrieval*, ser. SIGIR '08. New York, NY, USA: ACM, 2008. [Online]. Available: http://doi.acm.org/10.1145/1390334.1390435 pp. 587–594.

[27] B. Carterette, "On rank correlation and the distance between rankings," in *Proc. 32nd Int. SIGIR Conf. Research and Development in Information Retrieval*. ACM, 2009, pp. 436–443.

[28] R. Kumar and S. Vassilvitskii, "Generalized distances between rankings," in *Proc. 19th Int. World Wide Web Conf.*, Raleigh, NC, 2010, pp. 571–580.

[29] P. H. Lee and P. L. Yu, "Distance-based tree models for ranking data," *Computational Statistics and Data Analysis*, vol. 54, no. 6, pp. 1672 – 1682, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167947310000423

[30] Optify Inc., "The Changing Face of SERPs: Organic Click Through Rate [white paper]," http://www.optify.net/inbound-marketing-resources/new-study-how-the-new-face-of-serps-has-altered-the-ctr-curve, 2012.

[31] A. Tarsitano, "Comparing the effectiveness of rank correlation statistics," Università della Calabria, Dipartimento di Economia e Statistica, Tech. Rep., 2009.

[32] E. Tsiporkova and V. Boeva, "Fusing time series expression data through hybrid aggregation and hierarchical merge," *Bioinformatics*, vol. 24, no. 16, pp. i63–i69, 2008. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/24/16/i63.abstract

[33] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.-C. Tranchevent, B. De Moor, P. Marynen, B. Hassan et al., "Gene prioritization through genomic data fusion," *Nature biotechnology*, vol. 24, no. 5, pp. 537–544, 2006.

[34] W. J. Conover and R. L. Iman, "Rank transformations as a bridge between parametric and nonparametric statistics," *The American Statistician*, vol. 35, no. 3, pp. 124–129, 1981. [Online]. Available: http://www.jstor.org/stable/2683975

[35] A. Torsi, Y. Zhao, H. Liu, T. Tanzawa, A. Goda, P. Kalavade, and K. Parat, "A program disturb model and channel leakage current study for sub-20 nm nand flash cells," *IEEE Trans. Electron Devices*, vol. 58, no. 1, pp. 11–16, 2011.

[36] A. Barg and A. Mazumdar, "Codes in permutations and error correction for rank modulation," *IEEE Trans. Information Theory*, vol. 56, no. 7, pp. 3158–3165, July 2010.

[37] A. Mazumdar, A. Barg, and G. Zemor, "Constructions of rank modulation codes," in *Proc. IEEE Int. Symp. Information Theory*, July/Aug. 2011, pp. 869–873.

[38] P. Diaconis, *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics, 1988, vol. 11.

[39] L. Grupp, A. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. Siegel, and J. Wolf, "Characterizing flash memory: Anomalies, observations, and applications," in *42nd Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO-42)*, Dec. 2009, pp. 24–33.

[40] A. Cayley, "LXXVII. note on the theory of permutations," *Philosophical Magazine Series 3*, vol. 34, no. 232, pp. 527–529, 1849. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/14786444908646287

[41] D. B. West, *Combinatorial Mathematics*. Preliminary version, 2009.

[42] M. Deza and E. Deza, *Encyclopedia of distances*. Springer Verlag, 2009.

[43] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*. Cambridge, MA: MIT, 2001.

[44] V. I. Levenshtein, "On perfect codes in deletion and insertion metric," *Discrete Mathematics and Applications*, vol. 2, no. 3, pp. 241–258, 1992.

[45] W. D. Cook and M. Kress, "Ordinal ranking with intensity of preference," *Management Science*, vol. 31, no. 1, pp. 26–32, Jan. 1985.

[46] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," in *Proc. 10th Int. Conf. World Wide Web*. ACM, 2001, pp. 613–622.

[47] D. Sculley, "Rank aggregation for similar items," in *Proc. 7th SIAM Int. Conf. Data Mining*, Minneapolis, MN, 2007.

[48] F. Schalekamp and A. van Zuylen, "Rank aggregation: Together we're strong," *Proc. 11th Algorithm Engineering and Experiments (ALENEX)*, pp. 38–51, 2009.

[49] M. de Condorcet, *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: L'Imprimerie Royal, 1785.

[50] J.-C. de Borda, "Mémoire sur les élections au scrutin," *Histoire de l'Académie royale des sciences*, 1784.

[51] Economist Intelligence Unit, "A Summary of the Liveability Ranking and Overview [white paper]," http://www.eiu.com/site_info.asp?info_name=The_Global_Liveability_Report, Aug. 2012.

[52] American Marketing Association, "Dictionary." [Online]. Available: http://www.marketingpower.com/_layouts/Dictionary.aspx

[53] M. Sun, G. Lebanon, and K. Collins-Thompson, "Visualizing differences in web search algorithms using the expected weighted Hoeffding distance," in *Proc. 19th Int. Conf. World Wide Web*, Raleigh, NC, 2010. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772785 pp. 931–940.

[54] M. Deza and T. Huang, "Metrics on permutations, a survey," *J. Combinatorics, Information, and System Sciences*, vol. 23, pp. 173–185, 1998.

[55] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, July 1987.

[56] K. J. Arrow, *Social choice and individual values*. Yale Univ Pr, 1963.

[57] J. Hodge and R. Klima, *The mathematics of voting and elections: a hands-on approach*, ser. Mathematical World. American Mathematical Society, 2005, vol. 22.

[58] F. Farnoud, C.-Y. Chen, O. Milenkovic, and N. Kashyap, "A graphical model for computing the minimum cost transposition distance," in *Proc. IEEE Information Theory Workshop*, Dublin, Ireland, Aug./Sep. 2010.

[59] F. Farnoud and O. Milenkovic, "Decomposing permutations via cost-constrained transpositions," in *Proc. IEEE Int. Symp. Information Theory*, Saint Petersburg, Russia, July/Aug. 2011.

[60] F. Farnoud and O. Milenkovic, "Sorting of permutations by cost-constrained transpositions," *IEEE Trans. Information Theory*, vol. 58, pp. 3–23, Jan. 2012.

[61] N. Kashyap, *Personal Communication*, 2010.

[62] I. Goulden, "Tree-like properties of cycle factorizations," *J. Combinatorial Theory, Series A*, vol. 98, no. 1, pp. 106–117, Apr. 2002.

[63] D. Knuth, *The art of computer programming: Generating all combinations and partitions*. Addison-Wesley Professional, 2005.

[64] E. Slutsky, "Uber stochastische asymptoten und grenzwerte," *Metron*, vol. 5, 1925.

[65] F. Farnoud, B. Touri, and O. Milenkovic, "Nonuniform vote aggregation algorithms," in *Proc. Int. Conf. Signal Processing and Communications (SPCOM)*, Bangalore, India, July 2012.

[66] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Technical Report 1999-66, Nov. 1999, previous number = SIDL-WP-1999-0120. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[67] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/324133.324140

[68] H. P. Young, "An axiomatization of Borda's rule," *J. Econ. Theory*, vol. 9, no. 1, pp. 43–52, 1974.

[69] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, no. 3, pp. 228–236, Mar. 1965.

[70] J. Karlof, "Permutation codes for the Gaussian channel," *IEEE Trans. Information Theory*, vol. 35, no. 4, pp. 726–732, July 1989.

[71] I. F. Blake, G. Cohen, and M. Deza, "Coding with permutations," *Information and Control*, vol. 43, no. 1, pp. 1–19, 1979.

[72] C. J. Colbourn, T. Kløve, and A. C. H. Ling, "Permutation arrays for powerline communication and mutually orthogonal latin squares," *IEEE Trans. Information Theory*, vol. 50, no. 6, pp. 1289–1291, June 2004.

[73] J. Bruck, A. Jiang, and Z. Wang, "On the capacity of bounded rank modulation for flash memories," in *Proc. IEEE Int. Symp. Information Theory*, June/July 2009, pp. 1234–1238.

[74] F. Farnoud, V. Skachek, and O. Milenkovic, "Rank modulation for translocation error correction," in *Proc. IEEE Int. Symp. Information Theory*, July 2012, pp. 2988–2992.

[75] F. Farnoud, V. Skachek, and O. Milenkovic, "Error-correction in flash memories via codes in the ulam metric," *IEEE Trans. Information Theory*, vol. PP, no. 99, p. 1, 2013.

[76] A. Jiang, M. Schwartz, and J. Bruck, "Correcting charge-constrained errors in the rank-modulation scheme," *IEEE Trans. Information Theory*, vol. 56, no. 5, pp. 2112–2120, May 2010.

[77] I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," *IEEE Trans. Information Theory*, vol. 56, pp. 2551–2560, June 2010.

[78] A. Jiang and Y. Wang, "Rank modulation with multiplicity," in *Proc. IEEE GLOBECOM Workshops*, Dec. 2010, pp. 1866–1870.

[79] Z. Wang and J. Bruck, "Partial rank modulation for flash memories," in *Proc. IEEE Int. Symp. Information Theory*, June 2010, pp. 864–868.

[80] E. En Gad, M. Langberg, M. Schwartz, and J. Bruck, "Constant-weight Gray codes for local rank modulation," *IEEE Trans. Information Theory*, vol. 57, no. 11, pp. 7431–7442, Nov. 2011.

[81] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. New Jersey: Prentice Hall, Englewood Cliffs, 2004.

[82] D. Zhu and L. Wang, "On the complexity of unsigned translocation distance," *Theoretical Computer Science*, vol. 352, no. 1–3, pp. 322–328, 2006.

[83] M. Chou Kim, K. Oh, O. Chang Taek, S.-H. Choi, K. Belay, R. Elliman, and S. Russo, "Nonvolatile memories using deep traps formed in HfO2 by Nb ion implantation," *J. Applied Physics*, vol. 109, no. 5, 2011.

[84] G. Cellere, L. Larcher, A. Paccagnella, A. Visconti, and M. Bonanomi, "Radiation induced leakage current in floating gate memory cells," *IEEE Trans. Nuclear Science*, vol. 52, no. 6, pp. 2144 – 2152, 2005.

[85] P. Beame, E. Blais, and D. Huynh-Ngoc, "Longest common subsequences in sets of permutations," *Arxiv preprint arXiv:0904.1615*, 2009.

[86] A. Mazumdar, A. Barg, and G. Zemor, "Parameters of rank modulation codes: Examples," in *Proc. 49th Annu. Allerton Conf. Communication, Control, and Computing*, Sep. 2011, pp. 13–17.

[87] T. Kløve, T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Permutation arrays under the Chebyshev distance," *IEEE Trans. Information Theory*, vol. 56, no. 6, pp. 2611–2617, June 2010.

[88] T. Swart and H. Ferreira, "Decoding distance-preserving permutation codes for power-line communications," in *Proc. IEEE AFRICON*, Windhoek, South Africa, Sep. 2007, pp. 1–7.

[89] T. Wadayama and M. Hagiwara, "LP decodable permutation codes based on linearly constrained permutation matrices," in *Proc. IEEE Int. Symp. Information Theory*, Saint Petersburg, Russia, July/Aug. 2011, pp. 139–143.

[90] P. Frankl and M. Deza, "On the maximum number of permutations with given maximal or minimal distance," *J. Combinatorial Theory, Series A*, vol. 22, pp. 352–360, 1977.

[91] M. Deza and S. Vanstone, "Bounds for permutation arrays," *J. Statistical Planning and Inference*, vol. 2, no. 2, pp. 197–209, 1978.

[92] H. Tarnanen, "Upper bounds on permutation codes via linear programming," *European J. Combinatorics*, vol. 20, no. 1, pp. 101–114, 1999.

[93] A. A. Davydov, V. V. Zyablov, and R. E. Kalimullin, "Special sequences as subcodes of Reed-Solomon codes," *Problems of Information Transmission*, vol. 46, no. 4, pp. 321–345, Dec. 2010.

[94] J.-C. Chang, R.-J. Chen, T. Kløve, and S.-C. Tsai, "Distance-preserving mappings from binary vectors to permutations," *IEEE Trans. Information Theory*, vol. 49, no. 4, pp. 1054–1059, 2003.

[95] P. J. Cameron, "Permutation codes," *European J. Combinatorics*, vol. 31, no. 2, pp. 482–490, 2010.

[96] F.-W. Fu and T. Kløve, "Two constructions of permutation arrays," *IEEE Trans. Information Theory*, vol. 50, no. 5, pp. 881–883, May 2004.

[97] J.-C. Chang, R.-J. Chen, and S.-C. Tsai, "Distance preserving mappings from binary vectors to permutations," in *Proc. IEEE Int. Symp. Information Theory*, June/July 2003, p. 14.

[98] I. F. Blake, "Permutation codes for discrete channels (corresp.)," *IEEE Trans. Information Theory*, vol. 20, no. 1, pp. 138–140, Jan. 1974.

[99] J.-C. Chang, "Distance-increasing mappings from binary vectors to permutations that increase Hamming distances by at least two," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1683–1689, Apr. 2006.

[100] P. Neumann, "Encoding and decoding for cyclic permutation codes," *IRE Trans. Electronic Computers*, vol. EC-11, no. 4, pp. 507–511, Aug. 1962.

[101] J. Kim, "On increasing subsequences of random permutations," *J. Combinatorial Theory, Series A*, vol. 76, no. 1, pp. 148–155, 1996.

[102] J. Baik, P. Deift, and K. Johansson, "On the distribution of the length of the longest increasing subsequence of random permutations," *J. Amer. Math. Soc.*, vol. 12, pp. 1119–1178, 1999.

[103] D. Aldous and P. Diaconis, "Longest increasing subsequences: from patience sorting to the Baik-Deift-Johansson theorem," *Bulletin (new series) Amer. Math. Soc.*, vol. 36, pp. 413–432, 1999.

[104] A. M. Odlyzko and E. M. Rains, *On longest increasing subsequences in random permutations.* Providence, RI: American Mathematical Society, 2000, vol. 251, pp. 439–451.

[105] L. Su, F. Farnoud, and O. Milenkovic, "Computing the weighted transposition distance with star tree weights," in preparation.

[106] F. Raisali, F. Farnoud, and O. Milenkovic, "Weighted rank aggregation via relaxed integer programming," submitted to ISIT 2013.

[107] F. Farnoud, E. Yaakobi, O. Milenkovic, and J. Bruck, "Building consensus via iterative voting," submitted to ISIT 2013.

[108] M. Kim, F. Raisali, F. Farnoud, and O. Milenkovic, "Gene prioritization via rank aggregation with weighted distances," in preparation.